robotron

DCP

SOFTWARE
Dokumentation

Anleitung für den Assemblerprogrammierer
Teil II – Heft 1

Anleitung
fuer
den Assemblerprogrammierer
Teil 2 - Heft 1

VEB Robotron Buchungsmaschinenwerk
Karl-Marx-Stadt, 1987

Die vorliegende 1. Auflage der Dokumentation "Anleitung fuer den Assemblerprogrammierer" unter DCP 3.2 entspricht dem Stand vom 30.6.87 und unterliegt nicht dem Aenderungsdienst.

Nachdruck, jegliche Vervielfaeltigung oder Auszuege daraus sind unzulaessig.

Die Dokumentation wurde durch ein Kollektiv des

VEB Robotron Buchungsmaschinenwerk Karl-Marx-Stadt

erarbeitet.

Bitte senden Sie uns Ihre Hinweise, Kritiken, Wuensche oder Forderungen zur Dokumentation zu.

VEB Robotron Buchungsmaschinenwerk
Karl-Marx-Stadt
PSF 129
Karl-Marx-Stadt
9010

Die "Anleitung fuer den Assemblerprogrammierer" besteht aus zwei Teilen

Teil 1 enthaelt:

I.	CPU - Befehlsbeschreibung	Heft 1
II.	Assembler (MASM)	Heft 2

Teil 2 enthaelt:

III.	Editoren (EDLIN, BE)	Heft 1
IV.	Bibliotheksverwalter (LIB)	Heft 1
V.	Binder (LINK)	Heft 2
VI.	Debugger (SYMDEB)	Heft 2
VII.	MAKE	Heft 2

I N H A L T S V E R Z E I C H N I S

		Seite
III.	**EDITOREN**	6
1.	**Zeileneditor EDLIN**	6
1.1.	Einleitung	6
1.2.	Starten von EDLIN	6
1.3.	Allgemeine Kommandoinformationen	8
1.4.	EDLIN-Kommandos	12
1.4. 1.	A(ppend) - Nachlesen eines Teiles der editierten Datei	12
1.4. 2.	C(opy) - Kopieren von Zeilen	12
1.4. 3.	D(elete) - Loeschen von Zeilen	14
1.4. 4.	Zeile - Editieren einer Zeile	15
1.4. 5.	E(nd) - Ende des Editierens	16
1.4. 6.	I(nsert) - Einfuegen von Zeilen	17
1.4. 7.	L(ist) - Auflisten von Zeilen	17
1.4. 8.	M(ove) - Verschieben von Zeilen	18
1.4. 9.	P(age) - Seitenweises Anzeigen des Textes	19
1.4.10.	Q(uit) - Abbrechen des Editierens/Verwerfen der Datei	20
1.4.11.	R(eplace) - Textaustausch	21
1.4.12.	S(earch) - Textsuche	23
1.4.13.	T(ransfer) - Einfuegen einer Datei	24
1.4.14.	W(rite) - Speichern eines Teiles der editierten Datei	25
1.5.	Kommandozusammenfassung	26
2.	**Bildschirmeditor BE**	27
2.1.	Eigenschaften des Editors BE	27
2.2.	Starten des Editors BE	27
2.3.	Kommandobeschreibung	28
2.3. 1.	CHANGE - Suchen und Ersetzen von Text	29
2.3. 2.	DEFINE - Zuordnung von Funktionen zu den einzelnen Tasten	30
2.3. 3.	DIR - Anzeige des Inhaltsverzeichnisses	31
2.3. 4.	EDIT - Aufruf einer Datei zum Editieren	32
2.3. 5.	ERASE - Loeschen von Dateien	34
2.3. 6.	FILE - Abspeicherung einer Kopie der aktuellen Datei	34
2.3. 7.	LOCATE - Suchen einer Zeichenfolge	35
2.3. 8.	MACRO - Aufruf einer Kommandodatei	37
2.3. 9.	NAME - Aendern des aktuellen Dateinamens	38
2.3.10.	PRINT - Aufruf einer Datei zum Druck	39
2.3.11.	QUESTION MARK - Abruf der aktuellen Einstellungen fuer die Raender, die eingestellten Tabulatoren, zugeordnete Funktionen fuer eine Taste oder den verfuegbaren Hauptspeicherbereich	39
2.3.12.	QUIT - Editierung der aktuellen Datei beenden	41
2.3.13.	RENAME - Umbenennen einer Datei	41
2.3.14.	SAVE - Abspeicherung der aktuellen Datei	42
2.3.15.	SET DISPLAY - Setzen der Bildschirmparameter	43
2.3.16.	SET MARGINS - Setzen der Randeinstellungen	44

		Seite
2.3.17.	SET TABS - Setzen der Tabulatoren	45
2.4.	Funktionsbeschreibung	46
2.4. 1.	BACKTAB - Kursor auf vohergehende Tabulator-position	47
2.4. 2.	BACKTAB WORD - Kursor auf vorhergehendes Wort	47
2.4. 3.	BEGIN LINE - Kursor auf 1. Zeichen der Zeile	48
2.4. 4.	BEGIN MARK - Kursor an Beginn markierter Bereich	48
2.4. 5.	BOTTOM - Kursor auf letzte Zeile in der Datei	49
2.4. 6.	BOTTOM EDGE - Kursor auf letzte Zeile des Textbereiches	49
2.4. 7.	CENTER LINE - Zentrieren Zeile	50
2.4. 8.	COMMAND TOGGLE - Umschalten Editiermodus-Kommandomodus	50
2.4. 9.	CONFIRM CHANGE - Bestaetigung des Austausches bei Suchen mit Austauschen	51
2.4.10.	COPY MARK - Kopieren markierter Bereich	51
2.4.11.	CURSOR COMMAND - Kursor auf Befehlszeile	52
2.4.12.	CURSOR DATA - Kursor in den Datenbereich	53
2.4.13.	DELETE CHAR - Loeschen Zeichen	53
2.4.14.	DELETE LINE - Loeschen Zeile	53
2.4.15.	DELETE MARK - Loeschen eines markierten Bereiches	54
2.4.16.	DOWN - Kursor eine Zeile tiefer	55
2.4.17.	DOWN4 - Kursor um 4 Zeilen tiefer	55
2.4.18.	END LINE - Kursor an das Ende der Zeile	56
2.4.19.	END MARK - Kursor an das Ende eines markierten Bereiches	56
2.4.20.	ERASE BEGIN LINE - Loeschen bis Zeilenanfang	57
2.4.21.	ERASE END LINE - Loeschen bis Zeilenende	57
2.4.22.	ESCAPE - Eingabe von ASCII-Zeichen in den Text	57
2.4.23.	EXECUTE - Ausfuehrung eines Kommandos	58
2.4.24.	FILL MARK - Fuellen eines markierten Bereiches	59
2.4.25.	FIND BLANK LINE - Suche folgende Leerzeile	59
2.4.26.	FIRST NONBLANK - Suche erstes Textzeichen	60
2.4.27.	INDENT - Kursor an Beginn der Absatzeinrueckung	60
2.4.28.	INSERT LINE - Einfuegen einer Leerzeile	61
2.4.29.	INSERT MODE - Einfuegemodus einschalten	61
2.4.30.	INSERT TOGGLE - Ueberschreibemodus einschalten	62
2.4.31.	JOIN - Zwei Zeilen miteinander verbinden	62
2.4.32.	LEFT - Kursor ein Zeichen nach links	63
2.4.33.	LEFT8 - Kursor 8 Zeichen nach links	63
2.4.34.	LEFT40 - Kursor 40 Zeichen nach links	64
2.4.35.	LEFT EDGE - Kursor an das 1. Zeichen der Zeile	64
2.4.36.	LOWERCASE - Wandeln Gross- in Kleinbuchstaben	65
2.4.37.	MARK BLOCK - Einstellen Blockmarkierungen	65
2.4.38.	MARK CHAR - Einstellen Zeichenmarkierungen	66
2.4.39.	MARK LINE - Einstellen Zeilenmarkierungen	66
2.4.40.	MOVE MARK - Verschieben eines markierten Bereiches	67
2.4.41.	OVERLAY BLOCK - Ueberlagern durch einen Block	68
2.4.42.	PAGE DOWN - Blaettern eine Seite vorwaerts	68
2.4.43.	PGE UP - Blaettern eine Seite rueckwaerts	69
2.4.44.	REDRAW - Reorganisation Bildschirm nach Aenderung der Bildschirmparameter	69

		Seite
2.4.45.	REFLOW - Neuformatieren eines markierten Bereiches	70
2.4.46.	REPLACE MODE - Umschalten auf Ueberschreibemodus	70
2.4.47.	RIGHT - Kursor 1 Zeichen nach rechts	70
2.4.48.	RIGHT8 - Kursor 8 Zeichen nach rechts	71
2.4.49.	RIGHT40 - Kursor 40 Zeichen nach rechts	71
2.4.50.	RIGHT EDGE - Kursor zum rechten Rand	72
2.4.51.	RUBOUT - Loeschen Zeichen links vom Kursor	72
2.4.52.	SHIFT LEFT - Verschieben ein Zeichen nach links	73
2.4.53.	SHIFT RIGHT - Verschieben ein Zeichen nach rechts	73
2.4.54.	SPLIT - Aufteilen einer Zeile in zwei Zeilen	73
2.4.55.	TAB - Kursor zur naechsten Tabulatorposition	74
2.4.56.	TAB WORD - Kursor zum naechsten Wort	74
2.4.57.	TOP - Kursor in erste Zeile der Datei	75
2.4.58.	TOP EDGE - Kursor in erste Zeile des angezeigten Textbereiches	75
2.4.59.	UNDO - Rueckgaengigmachen einer Veraenderung	76
2.4.60.	UNMARK - Loeschen von Markierungen	76
2.4.61.	UP - Kursor 1 Zeile nach oben	77
2.4.62.	UPPERCASE - Wandeln Klein- in Grossbuchstaben	77
2.4.63.	UP4 - Kursor 4 Zeilen nach oben	78
2.5.	Standardtastaturzuweisung in BE.PRO	78
2.6.	Fehleranzeigen	85
Anhang	Uebersicht der Tastaturzuordnung fuer Kursorbewegungen und der Arbeit mit Markierungen	93
IV.	**BIBLIOTHEKSVERWALTER LIB**	94
1.	Einleitung	94
2.	Starten von LIB	95
2.1.	Starten mit Prompts	95
2.2.	Starten mit Befehlszeile	96
2.3.	Starten mit Antwortdatei	97
2.4.	Setzen Schalter "PAGESIZE"	98
3.	Schaffen einer neuen Bibliothek	99
4.	Pflege einer Bibliothek	99
4.1.	Hinzufuegen eines Moduls	100
4.2.	Loeschen eines Moduls	100
4.3.	Ersetzen eines Moduls	101
4.4.	Kopieren eines Moduls	101
4.5.	Transport eines Moduls	101
4.6.	Verbinden von Bibliotheken	102
5.	Pruefen des Inhaltes einer Bibliothek	102
6.	Erstellen einer Bibliotheks-Referenz-Datei	103
7.	Fehleranzeigen	104
8.	EXIT-Codes	107

III. EDITOREN

1. Zeileneditor EDLIN

1.1. Einleitung

Der Zeileneditor EDLIN dient zum Erzeugen, Aendern und Anzeigen von Dateien, gleichgueltig, ob es sich dabei um Quellprogramme oder Textdateien handelt.

EDLIN ist verwendbar zum:

- Erstellen und Speichern neuer Quelldateien
- Modifizieren existierender Dateien mit Abspeichern der originalen Datei.
- Loeschen, Aufbereiten, Einfuegen bzw. Anzeigen von Zeilen
- Suchen von Textstellen in einer oder in mehreren Zeilen und Loeschen oder Ersetzen dieser Textstelle
- Kopieren bzw. Loeschen von Zeilen.

Die maximale Editierzeilenlaenge betraegt 253 Zeichen.

Waehrend des Editiervorganges werden durch EDLIN Zeilennummern generiert und angezeigt, die lueckenlos aufsteigend aufeinander folgen. Diese sind jedoch kein Bestandteil der Quellzeilen in der gespeicherten Datei.

Beim Zeileneinfuegen erhoehen sich automatisch die Zeilennummern der nachfolgenden Zeilen um die Anzahl der eingefuegten Zeilen. Werden Zeilen geloescht, vermindert EDLIN automatisch die Zeilennummern der nachfolgenden Zeilen um die Anzahl der geloeschten Zeilen. Es entsteht eine lueckenlose Folge von Zeilennummern.

EDLIN speichert die bearbeiteten Dateien auf die aktuelle Diskette und das aktuelle Verzeichnis zurueck.

Alle in den nachfolgenden Punkten in eckige Klamern ([]) gesetzte Angaben sind wahlfreie Parameter.

1.2. Starten von EDLIN

EDLIN wird durch folgende Kommandoeingabe vom Betriebssystem aus gestartet:

EDLIN [<laufwerk>:][<pfad>]<dateibez.> [/B]

EDLIN ist das Kommando zum Aufruf des Editors.

<laufwerk>: ist die Laufwerkszuweisung, von dem eine Datei bereitgestellt werden soll.

<pfad> spezifiziert den Suchpfad fuer die Bereitstellung einer Datei.

<dateibez.> gibt den Dateinamen und die Dateierweiterung fuer die zu bearbeitende Datei an.

[/B] ist ein Wahlschalter, der besagt, dass das Dateiendekennzeichen (EOF) bei der Eingabedatei nicht beruecksichtigt werden soll. Seine Angabe ist bei Dateieinfuegungen notwendig, damit kein vorzeitiges Dateiende erkannt wird.

Soll eine neue Datei erzeugt werden, ist die <dateibez.> in jedem Fall einzugeben, um sie mit einem Namen zu versehen. Die **neue** Datei darf im **aktuellen** Verzeichnis noch **nicht** vorhanden sein. Findet EDLIN unter dem angegebenen Namen keine Datei, wird sie unter dem angegebenen Namen angelegt. Nachfolgend gibt EDLIN die Meldung aus:

Neue Datei
*_

Der Stern "*" ist das Aufforderungszeichen zur Kommandoeingabe.

Gestartet wird die Eingabe des Textes mit dem Kommando **I** (Einfuegen) und Druecken der Taste <ENTER>. Es erfolgt die Anzeige der aktuellen Zeilennummer (1) und des Aufforderungszeichens (*). Anschliessend kann der Text eingegeben werden.

Ist eine bereits existierende Datei zu editieren, muss ihre Dateibezeichnung beim Aufruf von EDLIN als Parameter mit angegeben werden. EDLIN laedt daraufhin die Datei in den Hauptspeicher. Kann die gesamte Datei geladen werden, erscheint folgende Meldung auf dem Bildschirm:

Ende der Eingabedatei
*_

Es ist nun die Datei mit Hilfe der EDLIN-Kommandos zu editieren. Passt die Datei aufgrund ihrer Groesse nicht vollstaendig in den Hauptspeicher laedt EDLIN soviele Zeilen, bis der Hauptspeicher zu 75% gefuellt ist. Danach erscheint das Eingabeaufforderungszeichen "*". Es kann nun der im Hauptspeicher befindliche Teil der Datei bearbeitet werden.

Zum Bearbeiten nachfolgender Teile der Datei muss zunaechst ein Teil der aufbereiteten Zeilen wieder in die Datei zurueckgeschrieben werden (mit dem WRITE-Kommando). Damit ist im Hauptspeicher Platz geschaffen. Danach sind mit dem APPEND-Kommando die noch nicht aufbereiteten Zeilen in den Hauptspeicher nachzuladen (siehe dazu die Kommandobeschreibung).

Am Ende des Editiervorganges werden die Originaldatei und die veraenderte Datei mit Hilfe des Kommandos **END** abgespeichert. Die

Originaldatei erhaelt dabei die Dateierweiterung **.BAK**. Die neue

Datei erhaelt die beim Aufruf von EDLIN unter <dateibez.> eingegebene Dateibezeichnung.

Eine urspruenglich bereits vorhandene .BAK-Datei wird durch EDLIN erst dann geloescht, wenn die neue Datei vollstaendig abgespeichert ist oder wenn der Platz auf der Diskette nicht zum vollstaendigen Abspeichern der neuen Datei ausreicht.

Es ist **nicht** moeglich, eine **.BAK-Datei** zu editieren. EDLIN betrachtet jede mit der Dateierweiterung .BAK versehene Datei als Sicherungsdatei. Soll eine solche Datei doch editiert werden, **muss** sie mit dem DCP-Kommando RENAME **umbenannt** werden. Danach kann diese mit dem neuen Namen versehene Datei zum Editieren aufgerufen werden.

1.3. Allgemeine Kommandoinformationen

EDLIN fuehrt die angegebenen Editierfunktionen ueber Textzeilen durch. Die folgenden Erlaeuterungen sollten bekannt sein, bevor die EDLIN-Kommandos verwendet werden. Kommandos koennen in Gross-, Kleinbuchstaben oder in gemischter Schreibweise eingegeben werden.

- **Pfadbezeichnungen** sind als Ergaenzung in EDLIN-Kommandos erlaubt. Mit dem folgenden Kommando wird beispielsweise die Datei TEXT.TXT des Verzeichnisses ANWEND/MUELLER editiert:

  EDLIN ANWEND\MUELLER\TEXT.TXT

- **Jede** Zeile kann durch eine Nummer, die relativ zur aktuellen Zeile (die mit einem "*" gekennzeichnet ist) liegt, angesprochen werden. Durch ein vorangestelltes **Minuszeichen** (-) wird eine Zeile vor der aktuellen Zeile und durch ein vorangestelltes **Pluszeichen** (+) eine Zeile nach der aktuellen Zeile adressiert.

  Beispiel:

  -10,+10L

  Dieses Kommando listet den Bereich ab 10 Zeilen vor der aktuellen Zeile bis 10 Zeilen nach der aktuellen Zeile auf dem Bildschirm auf.

- In **einer** Zeile koennen **mehrere** Kommandos nacheinander eingegeben werden. Sie muessen jedoch durch ein **Semikolon** (;) voneinander getrennt sein. Andernfalls wird die Fehlermeldung

  Eingabefehler

  angezeigt und das folgende Kommando ignoriert.

Beispiel:

15;5,+5L

Das erste Kommando setzt die aktuelle Zeile auf die Zeile 15. Nach Bedienung der <ENTER>-Taste wird das zweite Kommando ausgefuehrt. Dieses listet ab der Zeile 5 bis zur Zeile 20 den Text auf.

- Bei den Kommandos zum Suchen bzw. Ersetzen muessen die Zeichenfolgen mit ^Z (<CTRL>+<Z>) anstelle von <ENTER> beendet werden, falls die Zeile fortgesetzt werden soll. <ENTER> wird nur als Zeilenabschluss eingegeben.

Beispiel:

Sdiese Zeichenfolge<CTRL>+<Z>;-5,+5L

Das erste Kommando sucht "diese Zeichenfolge" ab der aktuellen Zeile +1 bis zum Dateiende und zeigt, wenn gefunden, diese Zeile an. Nach Bedienung der <ENTER>-Taste wird das zweite Kommando, die Anzeige des Textes 5 Zeilen vor der gefundenen Zeile bis 5 Zeilen danach, ausgefuehrt. Wird die Zeichenkette nicht gefunden, erfolgt die Meldung:

Nicht gefunden

Es erfolgt ebenso die Anzeige des Textes 5 Zeilen vor der weiterhin gueltigen aktuellen Zeile bis 5 Zeilen nach der aktuellen Zeile.

- EDLIN-Kommandos koennen **mit oder ohne** Leerzeichen zwischen einer Zeilennummer und dem Kommando eingegeben werden. So hat beispielsweise das Kommando 16D dieselbe Wirkung wie das Kommando 16 D. Es wird in jedem Falle die Zeile 16 geloescht.

- Im Einfuege-Modus koennen **Steuerzeichen** z.B. ^C (Tastenbedienung <CTRL>+<C>) eingegeben werden, indem das **Einleitungszeichen** ^V (Tastenbedienung <CTRL>+<V>) **vor** dem Steuerzeichen eingetastet wird. Dieses Einleitungszeichen gibt an, dass der **naechste** Grossbuchstabe ein Steuerzeichen ist. Steuerzeichen koennen ebenso in die Zeichenfolgen der Such- und Ersatz-Kommandos unter Verwendung dieses speziellen Einleitungszeichens mit eingegeben werden. Z.B.:

S<CTRL>+<V>Z

Dieses Kommando sucht das erste Auftreten von ^Z (<CTRL>+<Z>) in der Datei und zeigt diese Zeile an. Die Suche beginnt ab aktueller Zeile + 1.

R<CTRL>+<V>Z<CTRL>+<Z>ABC

Dieses Kommando sucht jedes Auftreten von ^Z (<CTRL>+<Z>) in der Datei und ersetzt dieses durch ABC.

R<CTRL>+<V>C<CTRL>+<Z><CTRL>+<V>Z

Dieses Kommando sucht jedes Auftreten von ^C (<CTRL>+<C>) und ersetzt dieses durch ^Z (<CTRL>+<Z>).
Um das Steuerzeichen ^V (<CTRL>+<V>) selbst in den Text einzuge ben, ist nachfolgende Eingabe vorzunehmen:

<CTRL>+<V><V>

- **^Z** (<CTRL>+<Z>) kennzeichnet in der Regel das **Dateiende.** Muss das Zeichen ^Z (<CTRL>+<Z>) an einer anderen Stelle in der Datei verwendet werden, ist anzugeben, dass es sich bei diesem Steuerzeichen nicht um das Ende der Datei handelt. Das wird beim Aufruf von EDLIN durch die Verwendung des **Schalters "/B"** erreicht. Durch die Angabe dieses Schalters wird die gesamte Datei unter Ignorieren aller in dieser Datei enthaltenen Dateiendekennzeichen zur Bearbeitung bereitgestellt.

- Wird die bearbeitete Datei zu gross, erfolgt die Meldung

  Arbeitsspeicher nicht ausreichend

  und es muss mit dem WRITE-Kommando ein Teil der Datei ausgelagert werden.

Kommandoparameter:

In einigen Kommandos koennen Parameter zur Spezifizierung des Kommandos angegeben werden. Die Wirkung der Parameter variiert in Abhaengigkeit davon, welches Kommando sie verwendet. Fuer einige Parameter sind Standardwerte voreingestellt, bzw. es wird der Wert des voherigen Aufrufes eines Kommandos verwendet.

Es gibt folgende Parameter:

<zeile> ist die Zeilennummer der zu bearbeitenden Zeile. Zeilennummern muessen durch ein Leerzeichen bzw. durch ein Komma (,) voneinander getrennt werden. Die Zeilennummern koennen in vier verschiedenen Arten angegeben werden:

1. als Zahl

Zahl ist eine beliebige Dezimalzahl, kleiner 65529. Wenn eine Zahl groesser als die hoechste in der Datei vorhandene Zeilennummer eingegeben wurde, stellt EDLIN die Zeile nach der letzten Zeile als Adressierung ein.

2. als Punkt

Wird anstelle einer Zahl ein Punkt (.) eingegeben, wird die aktuelle Zeile als Zeilenadresse angenommen. Die zuletzt editierte Zeile ist immer die aktuelle Zeile. Diese Zeile wird auf dem Bildschirm mit einem Stern

(*) gekennzeichnet. Dieser steht zwischen der angezeigten Zeilennummer und dem ersten Datenzeichen.
Bei Verwendung des Punktes fuer <zeile1> sind fuer <zeile2> nur positive relative Werte zugelassen.

3. als Nummernzeichen (#)

Das Nummernzeichen wird zur Adressierung der letzten Zeilennummer nach folgenden Zeilen verwendet. Es besitzt die gleiche Wirkung wie die Eingabe einer Zeilennummer, die groesser ist als die letzte in einer Datei.

4. mit <ENTER>

Wird die <ENTER>-Taste ohne Eingabe einer Zeilennummer bedient, adressiert sie die aktuelle Zeile.

Fragezeichen

Der Fragezeichenparameter eroeffnet den Dialog mit (?) EDLIN. Er wird in den Such- und Austauschkommandos verwendet (Search und Replace). Die Wirkung ist folgende:

Wird ein Kommando mit diesem Parameter abgearbeitet, wird die betreffende Zeile angezeigt. Danach fragt EDLIN durch die Ausschrift

O.K. ?

ob das Kommando weitergefuehrt werden soll.

Durch Eingabe von "J" oder <ENTER> wird das Kommando fortgesetzt. Jede andere Taste beendet das Kommando.

<zeichenfolge>

Dieser Parameter bezeichnet eine Zeichenfolge, die gesucht oder ersetzt bzw. die eine ersetzen soll. Zwischen dem Kommando und der Zeichenfolge bzw. zwischen den Zeichenfolgen duerfen keine Leerzeichen stehen, ausser, wenn diese zur Zeichenfolge gehoeren.

<n>

Der Parameter n ist eine Dezimalzahl im Wertebereich von 1 bis 65529. Er gibt eine Zeilenanzahl an.

<zahl>

Der Parameter <zahl> gibt die Anzahl der Wiederholungen eines Kommandos an. Es ist eine Dezimalzahl.

1.4. EDLIN-Kommandos

1.4.1. A(ppend) - Nachlesen eines Teiles der editierten Datei

Format

[<n>]A

<n> ist die Anzahl der nachzulesenden Zeilen.

A ist das Kommando.

Funktion:

Die im Parameter <n> angegebene Zeilenanzahl wird von der Diskettendatei gelesen und an das Ende der im Hauptspeicher befindlichen Datei angefuegt.

Anwendung:

Dieses Kommando darf nur zum Erreichen des kompletten Editierens sehr grosser Dateien (die nicht auf einmal in den Hauptspeicher eingelesen werden koennen), verwendet werden. EDLIN liest immer soviel wie moeglich Zeilen in den Hauptspeicher ein. Nur der der sich im Hauptspeicher befindliche Teil der Datei kann editiert werden. Zum Bearbeiten des Restes der Datei muss der gesamte bereits aufbereitete Text oder ein Teil davon mit dem WRITE-Kommando auf die Diskette zurueckgespeichert werden. Damit wird Platz im Hauptspeicher geschaffen. Erst danach kann der Rest oder ein weiterer Teil der Datei mit dem APPEND-Kommando eingelesen werden. Eine sehr grosse Datei wird folglich zum Editieren durch den Hauptspeicher "gerollt".

Wird Parameter <n> nicht eingegeben, laedt EDLIN den Hauptspeicher zu 75%. Ist der Hauptspeicher bereits zu 75% gefuellt, wird dieses Kommando ignoriert.

Wurde der gesamte Rest der Datei eingelesen, erfolgt die Meldung:

Ende der Eingabedatei

Ansonsten wird nur der "*" zur Eingabeaufforderung angezeigt.

1.4.2. C(opy) - Kopieren von Zeilen

Format

[<zeile1>],[<zeile2>],<zeile3>[,<zahl>]C

<zeile1> ist die Zeilennummer, ab der kopiert werden soll.

<zeile2> ist die Zeilennummer, bis zu der kopiert werden soll.

<zeile3> ist die Zeilennummer, vor die der Text zu kopieren (einzufuegen) ist.

<zahl> gibt an, wie oft der Text kopiert werden soll.

C ist das Kommando.

Funktion:

Dieses Kommando kopiert einen angegeben Bereich an eine ausgewaehlte Stelle in der Datei entsprechend des angegeben Wiederholungsfaktors.

Anwendung:

Dieses Kommando dient zum Vervielfaeltigen von Teilen der Datei. Ist <zeile1> nicht angegeben, wird ab der aktuellen Zeile bis zur <zeile2> kopiert.

Ist <zeile2> nicht angegeben, wird ab <zeile1> bis zur aktuellen Zeile kopiert.

Ist weder <zeile1> noch <zeile2> angegeben, wird nur die aktuelle Zeile kopiert.

Ist <zahl> nicht angegeben, wird nur einmal kopiert.

Nach dem Kopiervorgang wird die Datei automatisch neu durchnumeriert. Die Zeichen + und - sind nicht zugelassen.

Beispiele:

1,6,7C Es werden die Zeilen 1-6 vor die Zeile 7 kopiert.

8,11,#C Die Zeilen 8-11 werden an das Ende der Datei kopiert.

5,7,15,3C Es werden die Zeilen 5-7 3-mal vor die Zeile 15 kopiert.

,,10,2C Kopiert die aktuelle Zeile 2-mal vor die Zeile 10.

Wird keine Zielzeile eingegeben, erfolgt die Fehlermeldung:

Eingabefehler

Die Eingabe muss wiederholt werden.

1.4.3. D(elete) - Loeschen von Zeilen

Format

[<zeile1>][,<zeile2>]D

<zeile1> ist die Zeilennummer, ab der geloescht werden soll.

<zeile2> ist die Zeilennummer, bis zu der geloescht werden soll.

D ist das Kommando.

Funktion:

Dieses Kommando loescht einen angegebenen Bereich aus der Datei.

Anwendung:

Das Kommando wird angewendet, wenn aus der editierten Datei Zeilen geloescht werden muessen.

Wurde <zeile1> nicht angegeben, wird die Fehlermeldung

Eingabefehler

angezeigt. Die Eingabe ist zu wiederholen.

Ist <zeile2> nicht angegeben, wird nur die <zeile1> geloescht.

Ist <zeile1> und <zeile2> nicht angegeben, wird nur die aktuelle Zeile geloescht. Nach dem Loeschen wird die aktuelle Zeile auf die darauffolgende Zeile gesetzt und neu durchnumeriert.

Beispiele:

4D	Die Zeile 4 wird geloescht.
5,7D	Es werden die Zeilen 5-7 geloescht.
,9D	Fehlermeldung !
1,D	Die Zeile 1 wird nur geloescht.
D	Nur die aktuelle Zeile wird geloescht.
.,#D	Es wird ab der aktuellen Zeile bis zum Ende der Datei geloescht.

1.4.4. Zeile - Editieren einer Zeile

Format

[<zeile>]

<zeile> ist die Auswahl der zu editierenden Zeile.

Funktion:

Dieses Kommando stellt eine Zeile zum Editieren bereit.

Anwendung:

Das Kommando wird verwendet, wenn eine existierende Zeile in irgendeiner Weise veraendert werden soll. Mit diesem Kommando wird eine Zeile zum Editieren eroeffnet. Die Zeile wird mit der entsprechenden Zeilennummer auf dem Bildschirm angezeigt. Darunter erscheint die gleiche Zeilennummer mit dem Aufforderungszeichen (*) und der Rest ist leer. In diese Zeile sind die geaenderten Daten einzugeben. Die obere Zeile dient als Vorlage zum Editieren. Sie wird so lange festgehalten, bis das Editieren der Zeile mit <ENTER> abgeschlossen wird.

Zum Editieren stehen eine Reihe von Tasten zur Verfuegung, die eine bestimmte Funktion besitzen (siehe nachfolgende Tabelle).

Sollten keine Veraenderungen an der Zeile vorgenommen werden, ist nur <ENTER> zu bedienen. Diese Zeile wird in ihrem Originalzustand uebernommen, und die aktuelle Zeile wird auf die folgende Zeile eingestellt.

Wenn die <ENTER>-Taste bedient wird bevor die gesamte Zeile uebernommen wurde, werden die noch nicht uebernommenen Zeichen geloescht.

Soll eine editierte Zeile unveraendert bleiben, ist diese Zeile mit <ESC> oder <CTRL>+<Break> abzuschliessen.

Die nachfolgende Tabelle bezieht sich auf die zu editierende Zeile und nicht auf die Musterzeile.

Taste	Wirkung
< <-- >	Der Linkspfeil und Backspace loescht das zuletzt kopierte Zeichen.
<INS>	Diese Taste schaltet den INSERT-MODUS ein/aus. Sie ermoeglicht das Einfuegen von Zeichen in die Zeile.

Taste	Wirkung
< --> >	Der Rechtspfeil kopiert ein Zeichen aus der Musterzeile.
<DEL>	Die Bedienung dieser Taste bewirkt, dass ein Zeichen in der Musterzeile uebersprungen werden soll. Es dient zum Loeschen von Zeichen in der Zeile beim Editieren.
<F1>	Wirkt wie der Rechtspfeil.
<F2>+ <zeichen>	Diese Tastenkombination bewirkt das Kopieren von Zeichen ab aktueller Position bis vor das erste Auftreten des fuer <zeichen> eingegebenen Zeichens.
<F3>	Diese Taste kopiert den Rest der Zeile.
<F4>+ <zeichen>	Diese Tastenkombination bewirkt das Ueberspringen von Zeichen bis zum ersten Auftreten des fuer <zeichen> eingegebenen Zeichens.

1.4.5. E(nd) = Ende des Editierens

Format

E

Funktion:

Dieses Kommando beendet das Editieren und geht zurueck in das Betriebssystem.

Anwendung:

Das Kommando wird zum Abschluss der Editierung eingegeben. Es speichert die editierte Datei wieder zurueck auf die Diskette. Aus der Originaldatei wird eine Datei mit der Dateierweiterung .BAK. Diese Datei wird als Sicherungsdatei bezeichnet. Eine Datei mit der Dateierweiterung .BAK kann nicht zum Editieren aufgerufen werden. EDLIN laesst Sicherungsdateien zur Bearbeitung nicht zu.

Beim Erstellen einer neuen Datei wird keine .BAK-Datei angelegt. Soll eine .BAK-Datei editiert werden, muss vor dem Aufruf von EDLIN mit dem Betriebssystemkommando RENAME die Dateierweiterung von .BAK in eine andere umbenannt werden. Die Abspeicherung erfolgt immer auf die aktuelle Diskette in das aktuelle Verzeichnis!

1.4.6. I(nsert) - Einfuegen von Zeilen

Format

[<zeile>]I

<zeile>	Der Parameter <zeile> gibt an, vor welche Zeile der nachfolgend eingegebene Text eingefuegt werden soll.
I	ist das Kommando.

Funktion:

Dieses Kommando dient dem Einfuegen von Text in eine Datei.

Anwendung:

Ist eine neue Datei zu erzeugen bzw. etwas in eine bestehende Datei einzufuegen, ist dieses Kommando als erstes Kommando einzugeben und danach die Taste <ENTER> zu bedienen. Es erscheinen die Zeilennummer und der Stern (*) zur Eingabeaufforderung. Danach kann der Text eingegeben werden. Jede Zeile muss mit der <ENTER>-Taste abgeschlossen werden. Der Einfuegemodus wird durch Eingabe von <CTRL>+<Z> und nachfolgendem <ENTER> verlassen. Die aktuelle Zeile wird auf die naechstfolgende Zeile eingestellt und die Numerierung entsprechend angepasst.

Wenn <zeile> nicht mit eingegeben wurde, erfolgt das Einfuegen unmittelbar vor die aktuelle Zeile. Ist <zeile> eine Zeilennummer groesser als die letzte in der Datei, wird an das Ende der Datei angefuegt. In diesem Fall wird die letzte eingefuegte Zeile zur aktuellen Zeile.

1.4.7. L(ist) - Auflisten von Zeilen

Format

[<zeile1>][,<zeile2>]L

<zeile1>	ist die Zeile, ab der aufgelistet werden soll.
<zeile2>	ist die Zeile, bis zu der aufgelistet werden soll.
L	ist das Kommando.

Funktion:

Dieses Kommando listet den angegebenen Bereich der Datei auf dem Bildschirm auf (ab <zeile1> bis <zeile2>).

Anwendung:

Soll ein zusammenhaengender Teil der Datei aufgelistet werden, muss dieses Kommando eingegeben werden.

Wenn <zeile1> nicht mit eingegeben wurde, erfolgt die Auflistung 11 Zeilen vor der aktuellen Zeile bis zur <zeile2>.

Wurde <zeile2> nicht mit eingegeben, erfolgt die Auflistung ab der eingegeben <zeile1> in der Laenge von 23 Zeilen.

Ist <zeile1> und <zeile2> nicht mit eingegeben worden, werden insgesamt 23 Zeilen aufgelistet: 11 Zeilen vor und 11 Zeilen nach der aktuellen Zeile.

Beispiele:

L	Dieses Kommando listet 11 Zeilen vor der aktuellen Zeile bis 11 Zeilen nach der aktuellen Zeile auf.
15,28L	Die Zeilen 15 bis 28 werden auf dem Bildschirm angezeigt.
,28L	Das Kommando listet 11 Zeilen vor der aktuellen Zeile bis zur Zeile 28 auf dem Bildschirm auf.
15L	Mit diesem Kommando werden 23 Zeilen ab der Zeile 15 angezeigt.

1.4.8. M(ove) = Verschieben von Zeilen

Format

[<zeile1>],[<zeile2>],<zeile3>M

<zeile1>	ist die Anfangszeilennummer des Bereiches, der verschoben werden soll.
<zeile2>	ist die Endezeilennummer des Bereiches, der verschoben werden soll.
<zeile3>	ist die Zeile, vor die der verschobene Text eingefuegt wird.
M	ist das Kommando.

Funktion:

Dieses Kommando verschiebt einen angegebenen Textbereich innerhalb der Arbeitsdatei an eine andere Stelle.

Anwendung:

Das Kommando findet zum Umgruppieren von Zeilen in der Datei bzw. zum Aendern der Anordnungsreihenfolge ganzer Textteile Anwendung.

Wird <zeile1> nicht angegeben, wird ab der aktuellen Zeile bis zur <zeile2> verschoben.

Wurde <zeile2> nicht angegeben, wird ab <zeile1> bis zur aktuellen Zeile verschoben.

Wenn <zeile1> und <zeile2> nicht angegeben wurden, wird nur die aktuelle Zeile verschoben.

Nach erfolgter Verschiebung werden die Zeilen wieder neu durchnumeriert. Sollten sich die Zeilen, die verschoben werden sollen und die Zielzeile (<zeile3>) ueberlappen, bzw. <zeile3> wurde nicht mit eingegeben, erfolgt die Fehlermeldung:

Eingabefehler

Das Kommando muss erneut eingegeben werden.

Beispiele:

10,15,100M	Die Zeilen 10 - 15 werden vor die Zeile 100 verschoben.
,+25,40M	Es werden die Zeilen ab der aktuellen Zeile bis zur aktuellen Zeile + 25 vor die Zeile 40 verschoben.
7,,25M	Ab der Zeile 7 bis zur aktuellen Zeile werden alle Zeilen vor die Zeile 25 verschoben.
,,#M	Die aktuelle Zeile wird an das Dateiende verschoben.

1.4.9. P(age) - Seitenweises Anzeigen des Textes

Format

[<zeile1>**][**,<zeile2>**]P**

<zeile1>	ist die Zeile, ab der die seitenweise Anzeige erfolgen soll.
<zeile2>	ist die Zeile, die die seitenweise Anzeige abschliesst.
P	ist das Kommando.

Funktion:

Dieses Kommando fasst jeweils 23 Zeilen zu einer Seite zusammen und zeigt diese auf dem Bildschirm an.

Anwendung:

Dieses Kommando ist zur zusammenhaengenden seitenweisen Anzeige einer Datei auf dem Bildschirm zu verwenden.

Begonnen wird die Anzeige mit der durch <zeile1> festgelegten Zeile. Wurde <zeile1> nicht eingegeben, erfolgt die Anzeige ab der aktuellen Zeile und endet mit der durch <zeile2> festgelegten Zeile. Die aktuelle Zeile wird immer auf die letzte angezeigte Zeile gesetzt. Diese Zeile enthaelt auch den Stern zur Kennzeichnung. Das Weitergehen von der angezeigten Seite zur naechsten erfolgt durch Eingabe dieses Kommandos ohne Parameter.

Wurde <zeile2> nicht mit eingegeben, endet die seitenweise Anzeige mit Erreichen der aktuellen Zeile.

Beispiele:

15,#P	Es erfolgt eine seitenweise Anzeige von Zeile 15 bis zum Dateiende.
,45P	Ab der aktuellen Zeile bis zur Zeile 45 wird seitenweise angezeigt.
1,P	Es wird ab der Zeile 1 bis zur aktuellen Zeile seitenweise angezeigt.
P	23 Zeilen ab der aktuellen Zeile werden angezeigt. Es dient der Fortsetzung des Anzeigens.

1.4.10. Q(uit) = Abbrechen des Editierens/Verwerfen der Datei

Format

Q

Q	ist das Kommando.

Funktion:

Alle Veraenderungen an der Datei werden verworfen. Es erfolgt ein Abbruch des Editierens ohne Abspeichern der Datei.

Anwendung:

Dieses Kommando bricht das Editieren einer Datei ab. Es wird nichts auf die Diskette zurueckgespeichert. Die Originaldatei bleibt erhalten.

Nach Eingabe dieses Kommandos erfolgt die Anzeige:

Editieren abbrechen (J/N)?

Bei Eingabe von "J" wird abgebrochen. Es wird keine .BAK-Datei angelegt und nichts abgespeichert. Wenn "N" oder ein anderes Zeichen eingegeben wurde, wird mit dem Editieren fortgefahren.

1.4.11. R(eplace) = Textaustausch

Format

[<zeile1>][,<zeile2>][?]R<zeichenfolge1> <CTRL>+<Z>
[<zeichenfolge2>]

<zeile1>	ist die Zeilennummer, ab der die Suche und der Austausch erfolgen soll.
<zeile2>	ist die Zeilennummer, bis zu der die Suche und der Austausch erfolgen soll.
?	ist der Parameter fuer den Dialog mit dem Computer; Es wird jedesmal gefragt, ob der Austausch erfolgen soll oder nicht.
R	ist das Kommando.
<zeichenfolge1>	ist die zu suchende Zeichenfolge.
<CTRL>+<Z>	ist das Abschlusszeichen fuer die <zeichenfolge1>.
<zeichenfolge2>	ist die Zeichenfolge, die die <zeichenfolge1> ersetzen soll

Funktion:

Mit diesem Kommando koennen Textteile, Worte bzw. Zeichen durch anderen Text ausgetauscht werden.

Anwendung:

Dieses Kommando findet zum Suchen von Zeichenfolgen mit anschliessendem Zeichenfolgenersatz Anwendung. Sobald eine Zeichenfolge gefunden wird, erfolgt der Austausch. Jede Zeile, in der ein Ersatzvorgang durchgefuehrt wurde, wird auf dem Bildschirm angezeigt. Enthaelt eine Zeile mehrere Male <zeichenfolge1>, wird sie fuer jeden Austausch einmal angezeigt. Beendet ist dieses Kommando, wenn wieder der Stern (*) auf dem Bildschirm erscheint.

Soll <zeichenfolge2> eingegeben werden, ist die <zeichenfolge1> mit <CTRL>+<Z> abzuschliessen. Erst dann kann <zeichenfolge2> eingegeben werden. Wird <zeichenfolge2> nicht eingegeben, bedeutet das, dass <zeichenfolge1> durch eine leere Zeichenfolge ersetzt werden soll. Es wird also <zeichenfolge1> geloescht.

Wenn <zeichenfolge1> nicht mit eingegeben wird, verwendet EDLIN die bei einem vorherigen Such-/Ersatzvorgang angegebene Zeichenfolge. War es der erste Aufruf, wird das Kommando abgebrochen.

Wurde <zeile1> nicht angegeben, wird das Kommando ab aktueller Zeile + 1 begonnen.

Wenn <zeile2> nicht mit angegeben wurde, endet das Kommando am Dateiende (so, als waere das Nummernzeichen (#) als <zeile2> eingegeben worden).

Wurde Parameter "?" eingegeben, stoppt dieses Kommando an jeder die <zeichenfolge1> enthaltenden Zeile und zeigt die Aufforderung

O.K. ?

auf dem Bildschirm an.

Wird "J" oder <ENTER> eingegeben, wird <zeichenfolge1> durch <zeichenfolge2> ersetzt und das naechste Auftreten von <zeichenfolge1> gesucht. Beim Bedienen einer anderen Taste erfolgt kein Austausch. Dieser Zyklus wiederholt sich, bis das Ende des angegebenen Bereiches bzw. das Ende der Datei erreicht ist. Nach dem letzten Auftreten von <zeichenfolge1> zeigt EDLIN wieder den Stern (*) als Aufforderungszeichen fuer das naechste Kommando an. Durch Verwenden des Parameters "?" kann selbst entschieden werden, ob eine Zeichenfolge ausgetauscht werden soll oder nicht.

Beispiele:

1,8Rkette<ENTER>
"kette" wird in den Zeilen 1-8 gesucht und geloescht.

1,#Rheute<CTRL>+<Z>morgen<ENTER>
In der gesamten Datei wird "heute" durch "morgen" ersetzt.

,28?Rund<CTRL>+<Z>oder<ENTER>
Ab aktueller Zeile + 1 bis zur Zeile 28 wird das "und" durch "oder" im Dialog ersetzt.

R Wiederholen des vorherigen Austauschzyklusses ab aktueller Zeile bis zum Dateiende.

1.4.12. S(earch) - Textsuche

Format

[<zeile1>][,<zeile2>][?]S[<zeichenfolge1>]

<zeile1>	ist die Zeilennummer, ab der die Suche beginnen soll.
<zeile2>	ist die Zeilennummer bis wohin die Suche gehen soll.
?	Dieser Parameter gibt an, dass bei jeder gefundenen Zeichenkette gestoppt werden soll (Dialogarbeit).
S	ist das Suchkommando.
<zeichenfolge1>	ist die zu suchende Zeichenfolge.

Funktion:

Dieses Kommando sucht eine Zeichenfolge bzw. auch nur ein Zeichen in der Datei. Es wird die gefundene Zeile angezeigt.

Anwendung:

Immer dann, wenn eine bestimmte Stelle in der Datei gesucht werden soll, ist dieses Kommando anzuwenden.

Wurde <zeile1> nicht mit eingegeben, wird die Suche ab aktueller Zeile + 1 begonnen. Wenn <zeile2> nicht mit eingegeben wurde, wird bis zum Ende der Datei gesucht. Dies besitzt die gleiche Wirkung wie die Eingabe eines Nummernzeichen (#) fuer <zeile2>. Die zu suchende Zeichenfolge (<zeichenfolge1>) ist mit der <ENTER>-Taste abzuschliessen. Jede gefundene Zeile wird zur aktuellen Zeile.

Wenn <zeichenfolge1> nicht mit angegeben wurde, wird die in einem vorhergehenden Such- bzw. Austauschkommando als <zeichenfolge1> eingegebene Zeichenfolge verwendet. Wurde noch kein S-Kommando ausgefuehrt, wird abgebrochen.

Wenn kein Fragezeichen "?" mit eingegeben wurde, dann wird mit dem ersten Auftreten von <zeichenfolge1> das Kommando beendet und diese Zeile als aktuelle Zeile gekennzeichnet.

Wurde ein Fragezeichen eingegeben, stoppt das Programm an jeder Zeile, die <zeichenfolge1> enthaelt und es wird die Meldung

O.K. ?

angezeigt. Die Eingabe von "J" oder <ENTER> beendet das Kommando. Die gefundene Zeile wird zur aktuellen Zeile. Jede andere

Taste setzt den Suchvorgang fort. Wird <zeichenfolge1> nicht in der Datei oder dem angegebenen Bereich gefunden, erfolgt die Meldung auf dem Bildschirm:

Nicht gefunden

Beispiele:

5,15Soder<ENTER>
Im Bereich der Zeilen 5-15 wird das Auftreten der Zeichenfolge "oder" gesucht.

18S<CTRL>+<V>C<ENTER>
Das Steuerzeichen ^C (CTRL+C) wird in der Datei ab der Zeile 18 gesucht.

S<ENTER> Ab der aktuellen Zeile bis zum Dateiende wird eine Zeichenfolge gesucht, die in einem vorhergehenden Such- oder Austauschkommando verwendet wurde.

1.4.13. T(ransfer) - Einfuegen einer Datei

Format

[<zeile1>]T<dateibez.>

<zeile1> ist die Zeile, vor der die Datei eingefuegt werden soll.

T ist das Kommando.

<dateibez.> ist Laufwerk, Pfad, Dateiname, Dateierweiterung fuer die einzufuegende Datei.

Funktion:

Dieses Kommando fuegt eine Datei in die editierte Datei ein.

Anwendung:

Das Kommando kommt zur Anwendung, wenn aus verschiedenen Dateien eine neue Datei erzeugt werden soll, bzw. wenn bestimmte vorgefertigte Dateien in eine gerade bearbeitete Datei eingefuegt werden sollen.

Wurde <zeile1> nicht mit eingegeben wird die einzufuegende Datei vor der aktuellen Zeile eingefuegt. Die Datei wird anschliessend neu durchnumeriert. Die angegebene Datei wird immer im aktuellen Verzeichnis des aktuellen Laufwerkes bzw. in dem durch einen Pfad spezifizierten Verzeichnis beim Aufruf von EDLIN gesucht.

Beispiele:

15Ttext1.txt
Die Datei "texti.txt" wird vom aktuellen Laufwerk und vom aktuellen Verzeichnis vor die Zeile 15 in der Arbeitsdatei eingefuegt.

TB:anwen\mueller\text.txt
Es wird die Datei text.txt vom Laufwerk B: aus dem Verzeichnis ANWEN\MUELLER in die Arbeitsdatei eingefuegt.

1.4.14. W(rite) - Speichern eines Teiles der editierten Datei

Format

[<n>]W

<n> ist die Anzahl der Zeilen, die aus der Arbeitsdatei auf die Diskette zurueckgeschrieben werden sollen.

W ist das Kommando.

Funktion:

Dieses Kommando speichert Zeilen, die sich im Hauptspeicher zur Bearbeitung befinden zurueck auf die Diskette.

Anwendung:

Das Kommando wird nur zum Editieren sehr grosser Dateien, die nicht auf einmal in den Hauptspeicher eingelesen werden koennen, benoetigt. Dieses Kommando schafft einen Teil der bereits editierten Datei zurueck auf die Diskette. Damit wird zum Nachlesen des Restes der Datei mit dem Kommando APPEND Platz bereitgestellt.

Das Kommando ist ebenfalls zu benutzen, wenn beim Editieren der Datei mehr als 75% des Platzes des Hauptspeichers benoetigt wird.

Erfolgt keine Eingabe des Parameters <n> schreibt EDLIN so viele Zeilen auf die Diskette, bis eine Speicherfuellung von 70% erreicht ist.

Nach der Schreiboperation wird die Datei neu durchnumeriert. beginnend mit 1.

1.5. Kommandozusammenfassung

Kommando	Aufruf	Funktion
Zeile	<z1>	Zeile <z1> editieren
A	[<n>]A	n Zeilen nachlesen
C	[<z1>],[<z2>],<z3>[,<z>]C	Bereich kopieren
D	[<z1>][,<z2>]D	Bereich loeschen
E	E	Editierung beenden
I	[<z1>]I	Zeilen einfuegen
L	[<z1>][,<z2>]L	Text auflisten
M	[<z1>],[<z2>],<z3>M	Zeilen verschieben
P	[<z1>][,<z2>]P	Text durchblaettern
Q	Q	Abbruch der Editierung
R	[<z1>][,<z2>][?]R<zf1><CTRL>+<Z>	
	<zf2>	Zeichenfolgenersatz
S	[<z1>][,<z2>][?]S<zf1>	Zeichenfolgensuche
T	[<z1>]T<dateibez.>	Datei einfuegen
W	[<n>]W	Zeilen speichern

Zeichenerlaeuterung

<n> = eine Anzahl
<z1> = <zeile1>
<z2> = <zeile2>
<z3> = <zeile3>
? = Dialogparameter
<zf1> = <zeichenfolge1>
<zf2> = <zeichenfolge2>
<dateibez.> = <dateibezeichnung>

2. Bildschirmeditor BE

2.1. Eigenschaften des BE

Der Bildschirmeditor BE ist ein universeller Editor. Er wird zum Erstellen bzw. Aendern von Programm- bzw. Daten- oder Text-Dateien verwendet.

Hervorzuheben sind dabei folgende Moeglichkeiten des Editors:

- Randeinstellungen
- Setzen von Tabulatoren
- Verwendung von Markierungen
- Suchen, Loeschen, Kopieren und Verschieben von markierten Textteilen
- Suchen und Ersetzen von Zeichenfolgen automatisch bzw. selektiv
- Randanpassung, Wortumschlag und Neuformatierung des Textes
- Dateiausdruck
- Freie Funktionsbelegung der Tasten (fuer 99 Tasten)
- Ansteuerung von Farb- bzw. Schwarz/Weiss-Monitoren; Einstellung der Anzeigebreite (40 oder 80 Zeichen pro Zeile fuer Farbmonitore)
- Anzeige eines Hilfsmenues zur Unterstuetzung (Tastaturzuodnung fuer die wichtigsten Funktionen)
- Anzeige der Funktionszuordnung zu bestimmten Tasten
- Blockverschiebung
- Verzeichnisanzeige
- Gleichzeitiges Editieren von bis zu 20 Dateien (abhaengig vom Hauptspeicherplatz)
- Anwendung von Kommandodateien (Makro-Dateien)

Die Komponenten

BE.EXE
BE.PRO
BE.HLP

muessen sich beim Aufruf **im aktuellen Verzeichnis** befinden.

Dateien, die zur Bearbeitung aufgerufen werden, **koennen nicht durch einen Pfadnamen spezifiziert werden.** Es besteht nur die Moeglichkeit, einen Laufwerksnamen vor dem Dateinamen anzugeben. Ansonsten wird das aktuelle Verzeichnis verwendet.

Das Ergebnis der Arbeit wird in das aktuelle Verzeichnis, laut Aufruf, zurueckgeschrieben.

2.2. Starten des Editors BE

Der Editor wird vom Betriebssystem aus durch folgende Eingabe gestartet:

BE [<dateibezeichnung>]

jefolgt von <ENTER>. Wird schon beim Aufruf des Editors eine)ateibezeichnung mit angegeben, wird diese Datei zum Editieren ereitgestellt. Unter einer Dateibezeichnung ist der Dateiname, ein Punkt und die Dateierweiterung zu verstehen (in der Dokumentation abgekuerzt als <dateibez.>).

Es erfolgt die Anzeige des ersten Bildes. Dieses Bild muss mit <ENTER> quittiert werden. Es beinhaltet die Bezeichnung und die Version des Editors. Danach wird die Kommandodatei BE.PRO eingelesen und damit die Funktionszuordnung zu den Tasten vorgenommen.

Auf dem Bildschirm erscheint nun das Arbeitsbild und falls beim Aufruf eine Datei spezifiziert wurde, auch der Anfang der Datei. Wurde keine Datei zur Bearbeitung aufgerufen, erfolgt die Anzeige einer Leerdatei.

Jetzt kann mit der Editierung unter Verwendung der Editierkommandos und -funktionen begonnen werden.

Der Anfang der Datei wird durch

=== Dateianfang ===

markiert und das Dateiende durch

=== Dateiende ===.

Die letzten drei Zeilen auf dem Bildschirm beinhalten:

1. die Befehlszeile, auf der die Kommandos eingegeben werden muessen
2. die Statuszeile, in der die Dateibezeichnung fuer die aktuell in Bearbeitung befindliche Datei, die Zeilennummer und die Spaltennummer, in der sich der Kursor augenblicklich befindet und welche Eingabeart aktuell eingestellt ist (Einfuegen bzw. Ueberschreiben), angezeigt wird
3. die Ausgabe der Fehlermeldungen, falls Fehler erkannt werden bzw. Hinweise fuer die Weiterarbeit

2.3. Kommandobeschreibung

Die BE-Kommandos und Funktionen sind in alphabetischer Reihenfolge beschrieben. Fuer Funktionen, denen eine bestimmte Taste bzw. eine Tastenkombination zugeornet ist, wird diese in der Beschreibung mit angegeben.

Kommandos muessen in der Befehlszeile eingegeben werden. In fettgedruckte eckige Klammern (**[]**) eingeschlossene Parameter sind wahlfrei. Werden sie nicht eingegeben, wird ein Standardwert verwendet. Parameter sind in der Regel durch ein Leerzeichen voneinander zu trennen.

Kleingeschriebener Text in spitzen Klammern (<>) ist durch eine konkrete Angabe zu ersetzen. Grossgeschriebener Text in spitzen Klammern bedeutet eine Tastenbetaetigung. Schluesselwoerter werden in dieser Beschreibung gross geschrieben.

2.3.1. CHANGE - Suchen und Ersetzen von Text

Funktion:

Ein Zeichen bzw. eine Zeichenfolge wird gesucht und ersetzt. Das Ersetzen kann automatisch oder selektiv (mit Bedienerquittung) erfolgen.

Format

C<begrenzer><zeichenfolge1><begrenzer>
<zeichenfolge2><begrenzer>[-][*]

C	das auszufuehrende Kommando
<zeichenfolge1>	die zu suchende Zeichenfolge
<zeichenfolge2>	die zu ersetzende Zeichenfolge
-	Dieser Parameter legt die Arbeitsrichtung von der aktuellen Kursorposition zum Dateianfang fest. Wird dieser Parameter nicht angegeben, erfolgt die Ausfuehrung nach der aktuellen Kursorposition zum Dateiende hin.
*	Dieser Parameter bestimmt, dass das Kommando global, ohne zu fragen, durchgefuehrt werden soll. Wird der Parameter nicht mit eingegeben, wird selektiv gearbeitet.

Anwendung:

Dieses Kommando wird immer dann verwendet, wenn ein Zeichen oder eine Zeichenfolge in der Datei gesucht und ausgetauscht werden soll. Als Trennzeichen <begrenzer> fuer die Zeichenfolge koennen beliebige Zeichen (Sonderzeichen) verwendet werden. Sie muessen nur identisch innerhalb eines Kommando sein.

Die Zeichen in der Suchzeichenfolge <zeichenfolge1> koennen beliebig eingegeben werden (in Gross- oder Kleinbuchstaben oder in gemischter Schreibweise).

In der Ersatzzeichenfolge <zeichenfolge2> muessen die Zeichen so wie sie eingesetzt werden sollen geschrieben sein. Ist die <zeichenfolge2> nicht eingeben worden, wird der Suchbegriff durch einen Leerbegriff ersetzt; er wird geloescht.

Bei selektiver Arbeit wird der Kursor an die gefundene Zeichenfolge gesetzt und es muss nun entschieden werden, ob ausge-

tauscht werden soll oder nicht. Wenn ausgetauscht werden soll, muss die Funktion CONFIRM CHANGE aufgerufen werden (durch Druekken der Tasten <SHIFT>+<F5>).

Das Kommando kann durch die Eingabe von <CTRL>+<ENTER> wiederholt werden.

Beispiele:

1. **c(xyz(abc(**

Es wird "xyz" (in jeder moeglichen Schreibweise) selektiv ersetzt durch "abc", falls die Tasten <SHIFT>+<F5> gedrueckt werden. Mit <CTRL>+<ENTER> wird das Kommando fortgesetzt. Begonnen wird ab der aktuellen Kursorposition bis zum Dateiende.

2. **c/abc//***

Es wird "AbC" (in jeder moeglichen Schreibweise) gesucht und automatisch geloescht ab der aktuellen Kursorposition bis zum Dateiende.

3. **c!aBc!abc!-***

Es wird "abc" (in jeder moeglichen Schreibweise) ab der aktuellen Kursorposition zum Dateianfang hin gesucht und automatisch durch "abc" ersetzt.

4. **c'/'C'***

Es wird automatisch "/" ab der aktuellen Kursorposition bis zum Dateiende ersetzt durch "C".

Siehe auch das LOCATE-Kommando!

2.3.2. DEFINE - Zuordnung von Funktionen zu den einzelnen Tasten

Funktion:

Mit dem DEFINE-Kommando koennen den einzelnen Tasten bestimmte Editorfunktionen zugewiesen werden. Mit Bedienung dieser Tasten werden dann die entsprechenden Funktionen ausgefuehrt.

Format

DEF <taste> = ['<literal>'] [[<funktion>]]

DEF	das auszufuehrende Kommando
<taste>	der Tastenname
=	das Zuweisungszeichen

<literal> ein bereitgestellter Parameter

[<funktion>] der Name der auszufuehrenden Funktion

Anwendung:

Das Kommando wird zur Zuweisung der Editorkommandos bzw. -funktionen zu den einzelnen Tasten verwendet. Der Aufruf einer solchen Funktion erfolgt durch Betaetigen der entsprechend zugeordneten Taste.

Die Zuordnung der Funktionen zu den Tasten ist wahlfrei. Die Tastennamen sind reglementiert. So ist beispielsweise "s-f1" gleichzusetzen mit der Bedienung der Tasten <SHIFT>+<F1>. "ins" fuer die <INS>-Taste, "a-f1" fuer die Tasten <ALT>+<F1> bzw. "c-f1" fuer die Tasten <CTRL>+<F1> (siehe dazu Abschnitt 5.).

Den Tasten koennen Literale, Funktionen/Kommandos oder beides zugeordnet werden. Literale werden in einfache (') oder doppelte (") Hochkommas eingeschlossen.

Jede Tastendefinition ist auf einer Zeile fuer sich einzugeben. Die maximale Zeichenanzahl pro Zeile von 120 Zeichen darf nicht ueberschritten werden.

Die Tastendefinition wird in der Datei BE.PRO abgelegt. Diese kann mit dem Editor erfasst bzw. geaendert werden. Mit dem Aufruf und Start des Editors wird diese Datei eingelesen und interpretativ abgearbeitet. Diese Datei kann auch "MAKROS" enthalten. Mit dem Kommando e .keydefs koennen die aktuellen Tastenzuordnungen auf dem Bildschirm ausgeben bzw. geaendert werden (siehe Abschnitt 3.4.).

Beispiele:

1. **def del = [delete char]**

Loeschen eines Zeichens

2. **def a-f1 = [cursor command] set margins 1 80' [execute]**

Linken Rand auf Position 1 und rechten Rand auf Position 80 setzen

Siehe auch das MACRO-Kommando!

2.3.3. DIR - Anzeige des Inhaltsverzeichnisses

Funktion:

Alle Dateien des aktuellen Laufwerkes und des aktuellen Verzeichnisses, oder eine Gruppe von Dateien bzw. eine spezifizierte Datei werden auf dem Bildschirm aufgelistet.

Format

DIR [<laufwerk>:] [<dateibez.>]

DIR	das auszufuehrende Kommando
<laufwerk>:	Dieser Parameter bezeichnet das Laufwerk, von welchem das Inhaltsverzeichnis aufgelistet werden soll. Wird <laufwerk> nicht eingegeben, wird das aktuelle Laufwerk verwendet.
<dateibez.>	Dieser Parameter spezifiziert eine Datei bzw. eine Gruppe von Dateien. In einem Dateinamen sowie in der Dateierweiterung sind die Global-Sonderzeichen (? und *) zugelassen. Wird <dateibez.> nicht eingegeben, werden alle Dateien vom aktuellen Verzeichnis aufgelistet.

Anwendung:

Wenn ein Ueberblick ueber die vorhandenen Dateien auf einer Diskette benoetigt wird, ist dieses Kommando zu verwenden. Das angezeigte Verzeichnis wird in einer internen Datei (.DIR) abgelegt und kann bei Bedarf mit dem SAVE-Kommando abgespeichert bzw. mit dem RENAME-Kommando umbenannt werden.

Beispiele:

1. **dir b:test?**

Zeigt vom Laufwerk B: alle Dateien an, die mit test beginnen und denen ein beliebiges anderes Zeichen folgt.

2. **dir *.bas**

Zeigt vom aktuellen Laufwerk und vom aktuellen Verzeichnis alle Dateien an, die die Dateierweiterung .BAS tragen.

2.3.4. EDIT - Aufruf einer Datei zum Editieren

Funktion:

Eine Datei nach der anderen wird eroeffnet und zum Editieren bereitgestellt.

Format

E [<laufwerk>:][<dateibez.>] [NOTABS]

E	das auszufuehrende Kommando.

<laufwerk>: Dieser Parameter bezeichnet das Laufwerk, auf dem die Datei eroeffnet werden soll. Wird <laufwerk> nicht mit eingegeben, wird das aktuelle Laufwerk verwendet.

<dateibez.> Dieser Parameter spezifiziert eine Datei, die zum Editieren bereitgestellt werden soll. Wird sie nicht mit eingegeben, wird der aktuelle Dateiname verwendet.

NOTABS Durch Angabe dieses Parameters werden die eingegebenen Tabulatoren durch eine entsprechende Anzahl von Leerzeichen bei der Ausgabe auf Diskette ersetzt (standardmaessig stehen TAB-Stopp's auf den Positionen 9, 17, 25, 33 usw.). Wird dieser Parameter nicht mit eingegeben, bleiben Tabulatoren in der Datei erhalten. Es wird also die komprimierte Form abgespeichert.

Anwendung:

Dieses Kommando wird verwendet, wenn bereits eine Datei zum Editieren auf dem Bildschirm bereitgestellt wurde und zwischendurch eine andere Datei editiert werden soll. Der Teil der Datei, der auf dem Bildschirm steht, wird einschliesslich der Kursorposition und der Eingabeart (Einfuegen oder Ueberschreiben) gerettet und die neue aufgerufene Datei angezeigt bzw. neu eroeffnet. Nun kann diese Datei editiert werden. Mit der Bedienung der <F8>-Taste kann von der einen aktiven Datei zur anderen umgeschaltet werden, um dann mit dieser weiterzuarbeiten. Einmal eingestellte Raender und Tabulatoren sind fuer alle gleichzeitig zu bearbeitenden Dateien gueltig.

Beispiel:

Es ist bereits eine Datei (test1.txt) zur Bearbeitung auf dem Bildschirm, es soll aber zwischendurch die Datei test2.bas geaendert werden. Dabei ist folgendermassen zu verfahren:

Bedienen der <ESC>-Taste und Eingabe in der Kommandozeile:

e test2.bas

Quittieren dieser Eingabe mit der <ENTER>-Taste. Es wird nun die bereits auf dem Bildschirm angezeigte Datei test1.txt gerettet, die Datei test2.bas von der Diskette eingelesen und anschliessend auf dem Bildschirm angezeigt. Jetzt kann die Datei test2.bas veraendert werden. Wenn wieder zur Datei test1.txt zurueckgekehrt werden soll, ist die <F8>-Taste zu bedienen. Ein weiteres Druecken der <F8>-Taste schaltet auf die naechste Datei um.

2.3.5. ERASE - Loeschen von Dateien

Funktion:

Dateien werden von der Diskette geloescht.

Format

ERASE [<laufwerk>:]<dateibez.>

ERASE	das auszufuehrende Kommando
<laufwerk>:	Ist der Laufwerksname, auf dem geloescht werden soll. Wird der Parameter nicht eingegeben, wird das aktuelle Laufwerk verwendet.
<dateibez.>	Dieser Parameter spezifiziert eine Datei bzw. eine Gruppe von Dateien, die geloescht werden soll. In der <dateibez.> sind die Global-Sonderzeichen (? und *) zugelassen, sowohl im Dateinamen als auch in der Dateierweiterung.

Anwendung:

Dieses Kommando ist zu verwenden, wenn eine Datei nicht mehr benoetigt wird bzw. wenn Platz auf der Diskette geschaffen werden muss.

Beispiele:

1. **erase b:*.bas**

Es werden alle Dateien mit der Dateierweiterung .bas auf dem Laufwerk B: geloescht.

2. **erase Test?.***

Es werden vom aktuellen Laufwerk und dem aktuellen Verzeichnis alle Dateien, die im Dateinamen mit TEST beginnen und ein weiteres Zeichen enthalten und eine beliebige Dateierweiterung besitzen geloescht.

2.3.6. FILE - Abspeicherung einer Kopie der aktuellen Datei

Funktion:

Dieses Kommando sichert eine Kopie der aktuellen Datei vom Speicher auf die Diskette und schliesst diese Datei ab.

Format

FILE [<laufwerk>:][<dateibez.>] [NOTABS]

FILE	das auszufuehrende Kommando
<laufwerk>:	die Laufwerkszuordnung fuer diese Datei
<dateibez.>	Dieser Parameter spezifiziert eine der aktuellen Dateien, welche abgespeichert werden soll.
NOTABS	Dieser Parameter gibt an, dass die Datei unkomprimiert (mit Leerzeichen) ausgegeben werden soll.

Anwendung:

Wenn der momentane Arbeitsstand einer der aktuellen Dateien gerettet werden soll, ist dieses Kommando zu benutzen. Wurde nur das Kommando FILE und die <ENTER>-Taste in der Kommandozeile eingegeben, wird die gerade editierte Datei mit dem aktuellen Namen auf die Diskette zurueckgeschrieben (in komprimierter Form). Anschliessend wird zur naechsten Datei umgeschaltet bzw. wenn es die letzte Datei war, zurueckgegangen in das Betriebssystem. Wenn das Kommando FILE und eine <dateibez.> eingeben wurde, wird die ausgewaehlte Datei komprimiert abgespeichert und so verfahren, wie oben beschrieben. Wenn alle Parameter eingeben werden, wird die Datei unkomprimiert gespeichert.

Achtung!

Der Parameter NOTABS darf nur im Anschluss an eine Dateibez. eingegeben werden. Ansonsten wird die Datei NOTABS gesucht. Das file-Kommando darf **nicht** auf die internen Dateien (.KEYDEFS, .UNNAMED oder .DIR) angewendet werden. In die .KEYDEFS-Datei kann mit diesem Kommando kein Name eingefuegt werden

Beispiele:

Eingabe	Komprimierung	Dateinamens-aenderung
file	ja	nein
file <dateibez.>	ja	ja
file <dateibez.> NOTABS	nein	ja/nein

Siehe auch die Kommandos NAME und SAVE !

2.3.7. LOCATE - Suchen einer Zeichenfolge

Funktion:

Diese Funktion sucht das naechste bzw. vorhergehende Auftreten

einer Zeichenfolge und transportiert den Kursor an diese Stelle.

Format

L /<zeichenfolge>[/[-]]

L	das auszufuehrnde Kommando
/<zeichenfolge>	die zu suchende Zeichenfolge Der Schraegstrich (/) ist am Anfang zur Einleitung der Zeichenfolge erforderlich und am Ende zur Abgrenzung eines eventuell noch folgenden Zeichens, das nicht mehr zu der zu suchenden Zeichenfolge gehoert. Als Begrenzung der Zeichenfolge ist nur der Schraegstrich (/) zugelassen.
-	Das Minuszeichen (-) legt die Ausfuehrungsrichtung fuer das Suchen fest ab aktueller Kursorposition zum Dateianfang hin fest. Ansonsten wird zum Dateiende hin gesucht.

Anwendung:

Dieses Kommando ist zu verwenden, wenn schnell irgendeine bestimmte Zeichenfolge in der Datei gesucht werden soll, um von dort aus die Datei zu veraendern. Der Editor sucht diese Zeichenfolge, zeigt den Bereich, in dem sich die Zeichenfolge befindet, auf dem Bildschirm an und setzt den Kursor dorthin. Die Suchzeichenfolge kann gross, klein oder gemischt geschrieben werden, denn fuer die Suche sind nur die Zeichen massgebend und nicht ihre Schreibweise. So wird beispielsweise bei dem eingegebenen Suchbegriff /WaGeN in der Datei die Zeichenketten Wagen, wagen oder auch WAGEN gefunden.

Beispiele:

1. **l/abc**

Sucht vorwaerts das Auftreten von "abc" oder "ABC" als auch "aBc".

2. **l/abc/-**

Sucht denselben Begriff rueckwaerts.

3. **l'/abc'**

Sucht den Begriff "/abc" oder "/ABC" vorwaerts.

4. **l'/'-**

Sucht das Zeichen "/" rueckwaerts.

Siehe auch das CHANGE-Kommando !

2.3.8. MACRO - Aufruf einer Kommandodatei

Funktion:

Dieses Kommando liest eine Kommandodatei und stellt die dort angegebenen Kommandos bzw. Funktionen ein, die dann fuer die weitere Arbeit zur Verfuegung stehen (modifiziert bzw. ergaenzt die Einstellungen in der Datei BE.PRO).

Format

M [<laufwerk>:]<dateibez.>

M	das auszufuehrende Kommando
<laufwerk>:	Ist der Laufwerksname, von der die Datei eingelesen werden soll. Wird dieser Parameter nicht eingegeben, wird von der aktuellen Diskette und vom aktuellen Verzeichnis gelesen.
<dateibez.>	Dieser Parameter gibt den Dateinamen und die Dateierweiterung der zu lesenden Datei an.

Anwendung:

Dieses Kommando ist eine Ergaenzung zu der Datei BE.PRO. Mit diesem Kommando kann eine bestimmte, sich immer wiederholende Reihenfolge von Kommando- bzw. Funktionsaufrufen zusammengefasst und in einer sogenannten Makrodatei abgelegt werden. Diese wird aufgerufen, um die Zuordnungen in der Datei BE.PRO zu modifizieren bzw. zu ergaenzen.

Der Makroaufruf muss einer bestimmten Taste zugeordnet werden, da diese temporaer in die Datei BE.PRO eintragen wird. Die in einem Makro abzuarbeitenden Kommandos koennen bestimmte voreingestellte Werte ueberlagern. Sollte ein fehlerhaftes Kommando in einem Makro auftreten, erfolgt eine Fehlermeldung und die Abarbeitung wird an der Stelle, an der der Fehler aufgetreten ist, abgebrochen.

In der Makrodatei ist jedes Kommando auf einer Zeile fuer sich einzutragen. Die Vorschriften fuer die Erstellung des Makros sind im DEFINE-Kommando beschrieben.

Die Makros werden mit dem Editor erfasst. Durch Eingabe von edit .keydefs bzw. durch Druecken von <CTRL>+<F1> kann die aktuell eingestellte Funktionstastenbelegung auf den Bildschirm ausgegeben werden.

Beispiele:

Die folgenden drei Einstellungen sollen in der Makrodatei ueberschrieben werden. Nach dem Aufruf dieser Makrodatei werden diese Zuordnungen wirksam.

```
def a-f1 = [cursor command] 'set margins 1 80' [execute]
def ENTER = [insert line] [insert line]
def a-f2 = [cursor command] 's margins 10 70 15' [execute]
```

Mit dem ersten Kommando wird der Tastenbedienung <ALT>+<F1> die Einstellung des linken und rechten Randes auf die Werte von 1 und 80 zugeordnet.

Mit dem zweiten Kommando wird festgelegt, dass mit Bedienung der <ENTER>-Taste jeweils eine Zeile frei bleiben soll (zweizeilige Arbeit).

Das dritte Kommando stellt den linken Rand auf 10, den rechten Rand auf 70 und die erste Tabulator-Position auf den Wert 15 ein, falls die Tasten <ALT>+<F2> gedrueckt wurden.

2.3.9. NAME - Aendern des aktuellen Dateinamens

Funktion:

Die Dateibezeichnung der aktuell in Bearbeitung befindlichen Datei wird geaendert.

Format

```
N <dateibez.>
```

N	das auszufuehrende Kommando
<dateibez.>	Ist der neue Dateiname und die neue Dateierweiterung, die die alte Einstellung ersetzen sollen.

Anwendung:

Dieses Kommando wird verwendet, wenn die aufgerufene Datei nicht unter der aufgerufenen Dateibezeichnung wieder auf die Diskette zurueckgeschrieben werden soll, sondern unter einer neuen. Die Orginaldatei bleibt erhalten. Nach Ausfuehrung des Kommandos wird die neue Dateibezeichnung in der Statuszeile angezeigt. Ein nicht zugelassener oder fehlerhafter Dateiname verursacht die Fehlermeldung:

```
"Falscher Name"
```

auf dem Bildschirm.

Beispiel:

```
n test01.txt
```

Es wurde beispielsweise die Datei TEST1.TXT zum Editieren aufgerufen. Sie wird durch Eingabe des obigen Kommandos unter der Dateibezeichnung TEST01.TXT abspeichert.

2.3.10. PRINT - Aufruf einer Datei zum Druck

Funktion:

Das Kommando druckt die aktuelle Datei aus.

Format

```
P
```

P — das auszufuehrende Kommando

Anwendung:

Dieses Kommando ist zu verwenden, wenn eine Datei ausgedruckt werden soll. Der Druck kann durch Eingabe der Tastenfolge <CTRL>+<Num Lock> unterbrochen werden. Der Drucker stoppt nach der aktuellen Zeile. Durch Eingabe einer beliebigen anderen Taste wird der Druck fortgesetzt. Der Druck kann durch Eingabe der Tastenfolge <CTRL>+<Scroll Lock>+<Break> abgebrochen werden. Sollte der Drucker nicht bereit sein (nicht on-line, kein Papier eingelegt oder Drucker nicht eingeschaltet), so erfolgt die Fehlermeldung:

```
Kein Drucker; r/c (Wiederh./Abbruch)
```

auf dem Bildschirm. Der Fehler ist zu beseitigen und "r" einzugeben, wenn weitergedruckt werden soll oder "c" zum Stornieren des Kommandos. Die Datei wird vom Bildschirm aus gedruckt, einschliesslich Leerzeichen am Anfang und am Ende der Datei.

2.3.11. QUESTION_MARK - Abruf der aktuellen Einstellungen fuer die Raender, die eingestellten Tabulatoren, zugeordnete Funktionen fuer eine Taste oder den verfuegbaren Hauptspeicherbereich

Funktion:

Dieses Kommando zeigt die aktuell eingestellten Werte fuer Raender und Tabulatoren sowie die zugeordneten Funktionen der einzelnen Tasten und den verfuegbaren Hauptspeicherbereich an.

Format

```
? TABS
? MARGINS
? KEY <tastenname>
? MEMORY
```

?	das auszufuehrende Kommando
TABS	das Schluesselwort fuer die Tabulatoren, deren Einstellung angezeigt werden soll
MARGINS	das Schluesselwort fuer die Raender, deren Einstellung angezeigt werden soll
KEY	das Schluesselwort fuer die Anzeige der zugeordneten Funktionen zu einer Taste
<tastenname>	reglementierter Name der Taste
MEMORY	das Schluesselwort fuer die Anzeige des verfuegbaren Speichers in Bytes

Anwendung:

Die aktuell eingestellten Werte koennen mit Hilfe dieses Kommandos abgerufen werden. Nach Eingabe von ? TABS auf der Kommandozeile und Druecken der <ENTER>-Taste wird in der Kommandozeile die aktuelle Einstellung der Tabulatoren angezeigt. Der Kursor steht am Anfang dieser Zeile. Die angezeigten Werte koennen an den angegebenen Positionen ueberschrieben werden. Nach Bedienung der <ENTER>-Taste wird die neue Einstellung uebernommen und es kann das naechste Kommando eingeben werden.

Das hier genannte trifft ebenfalls auf die Eingabe von ? MARGINS zu, nur dass hier die Einstellung der Raender und die Einruekkung fuer Absaetze erfolgt.

Bei der Eingabe von ? KEY ENTER beispielsweise erfolgt die Anzeige der Definition der Funktionen fuer die <ENTER>-Taste in der Kommandozeile. Diese Zuordnung kann nicht veraendert werden. Das gleiche gilt fuer die Eingabe ? MEMORY, nur dass hier der verfuegbare Speicherbereich in der Hinweiszeile erscheint.

Beispiele:

Eingabe	Anzeige
? TABS	set tabs 1 9 17 25 33 41 49 ...
? MARGINS	set margins 1 254 1
? KEY ENTER	def ENTER = [insert line]
? MEMORY	315664 Bytes frei

2.3.12. QUIT - Editieren der aktuellen Datei beenden

Funktion:

Das Editieren wird beendet und die editierte Datei aus dem Speicher entfernt.

Format

```
Q
```

Q — das auszufuehrende Kommando

Anwendung:

Dieses Kommando ist zu benutzen, wenn das Editieren einer Datei ohne Rueckspeicherung beendet werden soll (Verwerfen der Datei). Nach dem Eingeben dieses Kommando erscheint in der Statuszeile auf dem Bildschirm die Ausschrift:

```
Abbruch ? y/n
```

Diese Anzeige erfolgt nur, wenn etwas an der Datei veraendert wurde. Durch Druecken von "n" (nein) wird das Kommando ignoriert und es kann weitergearbeitet werden. Durch Eingabe von "y" (ja) wird die aktuelle Datei verworfen und auf die vorhergehende eroeffnete Datei umgeschaltet. Sollte keine weitere vorhanden sein, wird der Editor verlassen und es erfolgt der Ruecksprung ins Betriebssystem.

2.3.13. RENAME - Umbenennen einer Datei

Funktion:

Dateien werden auf einer Diskette umbenannt.

Format

```
RENAME [<laufwerk>:]<alte dateibez.> <neue dateibez.>
```

RENAME	das auszufuehrende Kommando
<laufwerk>:	Dieser Parameter spezifiziert das zu verwendende Laufwerk.
<alte dateibez.>	Dieser Parameter spezifiziert die Datei, die umbenannt werden soll.
<neue dateibez.>	Dieser Parameter spezifiziert den neuen Dateinamen.

Anwendung:

Dieses Kommando wird benutzt um Dateien umzubenennen. Der neue Dateiname darf im Verzeichnis noch nicht enthalten sein. Er muss verschieden von bereits eroeffneten Dateien sein. In den Namensangaben der Dateien sind die Global-Sonderzeichen (? und *) zugelassen (Dateiname und Dateierweiterung).

Wird ein Dateiname nicht gefunden, erfolgt in der Hinweiszeile die Fehlermeldung:

Datei nicht gefunden

Beispiele:

1. **rename test?.txt *.doc**

Es werden ayf dem aktuellen Laufwerk alle Dateien, deren Dateinamen aus TEST und einem beliebigen Zeichen bestehen und die die Dateierweiterung .txt tragen, in Dateien mit der Dateierweiterung .doc umbenannt.

2. **rename b:test.doc fert.***

Es wird die Datei test.doc auf dem Laufwerk B: umbenannt in fert.doc.

2.3.14. SAVE - Abspeicherung der aktuellen Datei

Funktion:

Dieses Kommando rettet eine Kopie der aktuellen Datei auf die Diskette (Zwischenretten einer Datei).

Format

SAVE [<laufwerk>:][<dateibez.>] [NOTABS]

SAVE	das auszufuehrende Kommando.
<laufwerk>:	Ist der Laufwerksname, auf dem die Kopie der aktuellen Datei abgespeichert werden soll. Wenn nicht angegeben, erfolgt die Ausgabe auf das aktuelle Laufwerk.
<dateibez.>	Ist der Dateiname und die Dateierweiterung unter dem die Datei abgespeichert werden soll. Wenn nicht angegeben, wird unter der aktuellen Dateibezeichnung abgespeichert.
NOTABS	Dieser Parameter bestimmt, dass die Datei unkomprimiert ausgegeben werden soll (mit allen Leerzeichenfolgen). Wenn nicht angegeben, erfolgt die Ausgabe komprimiert.

Anwendung:

Dieses Kommando wird zum Retten des momentanen Arbeitsstandes der aktuellen Datei benutzt. Die aktuelle Datei wird nicht aus dem Speicher entfernt. Sie kann weiterbearbeitet werden. Das Kommando ist nicht auf die internen Dateien (.KEYDEFS, .UNNAMED und .DIR) anwendbar. Diese Dateien muessen erst umbenannt werden. Danach koennen sie wie gewoehnliche Dateien abgespeichert werden.

Beispiele:

Eingabe	Komprimiert	Namensaenderung
save	ja	nein
save <dateibez.>	ja	ja
save <dateibez.> NOTABS	nein	ja/nein

Siehe auch das FILE-Kommando!

2.3.15. SET DISPLAY - Setzen der Bildschirmparameter

Funktion:

Dieses Kommando aendert die Anzeigeart der aktuellen Datei auf dem Bildschirm entsprechend der eingegebenen Parameter.

Format

```
S DISPLAY MONO       (fuer Monochrombildschirm)
S DISPLAY COLOR 40   (40 Zeichen/Zeile farbig)
S DISPLAY COLOR 80   (80 Zeichen/Zeile farbig)
S DISPLAY B/W 40     (40 Zeichen/Zeile schwarz/weiss)
S DISPLAY B/W 80     (80 Zeichen/Zeile schwarz/weiss)
```

S DISPLAY	das auszufuehrende Kommando
MONO	der Parameter fuer Monochrombildschirm 80 Zeichen/Zeile
COLOR	der Parameter Farbmonitor
B/W	der Parameter fuer schwarz/weiss
40 bzw.80	der Parameter fuer Zeichen/Zeile

Anwendung:

Dieses Kommando wird verwendet, wenn die Darstellungsart auf dem Bildschirm und die Anzahl Zeichen/Zeile veraendert werden sollen. Die folgende Tabelle gibt die Zuordnung der Wirkungen fuer die einzelnen Parameter an.

Parameter	mono	color 40/80	b/w 40/80
Daten-bereich	weiss auf schwarz	weiss auf schwarz	weiss auf schwarz
Kommando-Zeile	schwarz auf weiss	blau auf weiss	schwarz auf weiss
Status-Zeile	weiss auf schwarz	weiss auf schwarz	weiss auf schwarz
Hinweis-Zeile	intensiv weiss	rot auf schwarz	weiss auf schwarz
markierter Bereich	weiss auf schwarz	weiss auf schwarz	weiss auf schwarz
Kursor-Position	intensiv unterstrichen	blau auf schwarz	blinkend weiss auf schwarz
markierte Position	schwarz unterstrichen	blau auf weiss	blinkend schwarz auf weiss

2.3.16. SET MARGINS - Setzen der Randeinstellungen

Funktion:

Dieses Kommando stellt den linken und den rechten Rand sowie die Einrueckticfe fuer Absaetze auf einen bestimmten Wert ein.

Format

S MARGINS <n1> <n2> [<n3>]

S MARGINS — das auszufuehrende Kommando

<n1> — die Position des linken Randes

<n2> — die Position des rechten Randes

<n3> — die Einrueckticfe eines Absatzes; Standardwert ist der linke Rand.

Anwendung:

Dieses Kommando legt die Grenzen der Texteingabe in der Zeile fest.

Bei der Eingabe der Raender muss der rechte Rand groesser sein als der linke.

Die Absatzeinrueckung kann kleiner sein als der linke Rand, wenn sich dieser nicht auf der Position 1 befindet.

Standardwerte sind:

- linker Rand Position 1
- rechter Rand Position 254
- Absatzeinrueckung Position 1

Soll nicht mit diesen voreingestellten Werten gearbeitet werden, sind diese Werte mit diesem Kommando neu festzulegen. Die eingestellten Werte bleiben so lange erhalten, bis der Editor verlassen wird. Die Randpositionen werden nicht in der Datei gespeichert. Sollen andere Standardwerte fuer die Randeinstellungen verwendet werden, muessen diese in der Datei BE.PRO geaendert werden bzw. es ist ein entsprechender Makro zu schaffen.

Der Editor besitzt einen sogenannten Wortumschlag, d.h., wenn ein Wort den rechten Rand ueberschreitet, wird es auf den Anfang der folgenden Zeile uebernommen.

Beispiele:

Den linken Rand auf den Wert 5, den rechten Rand auf den Wert 62 und die Absatzeinrueckung auf den Wert 10 setzen:

1. **def s-f10 = 's margins 5 62 10' [execute]**

Mit dieser Makrozuordnung in der Datei BE.PRO wird der Taste <SHIFT>+<F10> diese Funktion zugeordnet.

2. **set margins 5 62 10**

Diese Einstellung in der Kommandozeile ersetzt die entsprechenden Werte aus der Datei BE.PRO.

Siehe dazu auch die Kommandos DEFINE und MACRO!

2.3.17. SET TABS - Setzen der Tabulatoren

Funktion:

Dieses Kommando dient der Einstellung der Tabulatorpositionen, die mittels der Tabulatortaste angesprungen werden koennen.

Format

S TABS <t1> <t2> <t3> ... <t20>

S TABS	das auszufuehrende Kommando
<t1> - <t20>	die einzelnen Tabulatorpositionen

Anwendung:

Mit diesem Kommando koennen die Standardtabulatoren veraendert werden. Standardmaessig ist aller 8 Spalten ein Tabulator gesetzt (9, 17, 25 ...).

Maximal 20 Tabulatoren im Bereich von 1 bis 254 koennen eingestellt werden. Die Einstellungen muessen aufsteigend aufeinander folgen.

Durch Bedienen der <TAB>-Taste wird der Kursor nach rechts zur naechsten Tabulatorposition transportiert und durch Druecken von <SHIFT>+<TAB> wird er nach links an die vorhergehende Tabulatorposition gesetzt. Dieser Tastenbedienung sind die Funktionen [TAB] bzw. [BACKTAB] in der Datei BE.PRO zugewiesen. Die aktuell eingestellten Tabulatoren koennen mit dem Kommando QUESTION MARK durch Eingabe von ? TABS auf dem Bildschirm dargestellt werden.

Siehe auch die Funktionen [TAB], [BACKTAB], [TAB WORD], [BACKTAB WORD] und das Kommando QUESTION MARK!

2.4. Funktionsbeschreibung

Die Funktionen sind in alphabetischer Reihenfolge aufgefuehrt. Die voreingestellten Werte fuer ein Kommando und die zugeordnete Taste fuer eine Funktion sind in der Datei BE.PRO festgelegt. Sollten andere Funktionen oder Werte benoetigt werden als die standardmaessig zugeordneten, ist diese Datei zu veraendern. Das gleiche gilt fuer die zugeordneten Tasten. Die Tastennamen sind reglementiert.

Fuer Doppeltastenbedienung gelten folgende Zuordnungen:

- fuer die Sondertaste <SHIFT> (Umschalten Klein/Gross) ist ein "s" vorangestellt
- fuer die <CTRL>-Taste ein "c"
- fuer die <ALT>-Taste ein "a"

Siehe das DEFINE-Kommando!

Bei der Zuordnung der Funktionen muss der Funktionswert in eckigen Klammern ([]) angeben werden.

Funktionen koennen nicht wie die Kommandos auf der Befehlszeile eingegeben werden. Die zu verwendenden Funktionen muessen unbedingt Tasten zugeordnet sein, bevor sie aufgerufen werden koen-

nen.

2.4.1. BACKTAB - Kursor auf vorhergehende Tabulatorposition

Funktion:

Diese Funktion transportiert den Kursor nach links zur vorhergehenden Tabulatorposition.

Format [BACKTAB]

Taste <SHIFT>+<TAB>

Anwendung:

Tabulatorpositionen sind standardmaessig aller 8 Positionen eingestellt beginnend ab der Position 1 in der Folge 9, 17, 25, 33, 41 ....

Mit dem SET TABS-Kommando kann diese Einstellung veraendert werden. Die Einstellung der aktuellen Tabulatoren kann mit dem Kommando QUESTION MARK (? TABS) abgerufen werden.

Ist die erste eingestellte Tabulatorposition bzw. der linke Rand erreicht, erfolgt keine weitere Positionierung.

Siehe auch die Funktionen [TAB], [BACKTAB WORD] und die Kommandos SET TABS und QUESTION MARK!

2.4.2. BACKTAB WORD - Kursor auf vorhergehendes Wort

Funktion:

Diese Funktion transportiert den Kursor nach links unter das erste Zeichen eines Wortes.

Format [BACKTAB WORD]

Taste keine Zuordnung

Anwendung:

Ein Wort ist definiert durch eine Folge von Zeichen, die in Leerzeichen eingeschlossen sind. Der Kursor wird auf das erste Zeichen, das nach einem Leerzeichen steht, positioniert bzw. an das erste Zeichen der Zeile, wenn diese leer ist.

Ist die Zeile laenger als die eingestellte Bildschirmbreite und wird der Bildschirmrand durch die Positionierung ueberschritten, wird das angezeigte Bild um 20 bzw. 40 Zeichen, je nach Einstellung der Bildschirmbreite, horizontal nach rechts bzw. links gerollt.

Siehe auch die Funktionen [TAB], [TAB'WORD] und [BACKTAB]!

2.4.3. BEGIN LINE - Kursor auf 1. Zeichen der Zeile

Funktion:

Diese Funktion transportiert den Kursor auf die erste Position der Zeile.

Format **[BEGIN LINE]**

Taste <HOME>

Anwendung:

Mit dieser Funktion wird der Kursor schnell an die erste Position der Zeile positioniert. Ist der Kursor auf einer hoeheren Stelle als der Position 40 bzw. 80, je nach eingestellter Bildschirmbreite, wird der Anzeigebereich ab der Position 1 neu angezeigt. Sollte sich der Kursor bereits in der Position 1 befinden, erfolgt keine Positionierung.

Siehe auch die Funktion [END LINE] !

2.4.4. BEGIN MARK - Kursor an Beginn markierter Bereich

Funktion:

Der Kursor wird unter das erste Zeichen eines markierten Bereiches transportiert.

Format **[BEGIN MARK]**

Taste <ALT>+<Y>

Anwendung:

Diese Funktion positioniert den Kursor schnell an den Beginn einer markierten Stelle (unter das erste Zeichen) bei einer Block- oder Zeichenmarkierung.

Bei einer Zeilenmarkierung wird der Kursor nur vertikal zur entsprechenden Zeile transportiert, ohne die Spaltenposition zu veraendern.

Markierungen haben **nur** fuer die aktuellen Dateien Gueltigkeit. Es kann in einer Datei nur **eine** Stelle markiert werden. Markierungen werden nicht mit auf die Diskette abgespeichert.

Wird diese Funktion verwendet, ohne dass eine Markierung vorhanden ist, erscheint in der Hinweiszeile die Fehlermeldung:

Markierung fehlt

2.4.5. BOTTOM - Kursor auf letzte Zeile in der Datei

Funktion:

Diese Funktion transportiert den Kursor auf die letzte Zeile der Datei.

Format **[BOTTOM]**

Taste <CTRL>+<END>

Anwendung:

Beim Betaetigen der Tastenkombination <CTRL>+<END> wird der Kursor schnell auf die letzte Zeile der aktuellen Datei transportiert.

Sollte sich der Kursor in der Befehlszeile befinden und es werden diese Tasten gedrueckt, erfolgt die gleiche Wirkung, d.h., der Kursor steht dann auf der letzten Zeile der aktuellen Datei.

Siehe auch die Funktion [TOP]!

2.4.6. BOTTOM EDGE - Kursor auf letzte Zeile des Textbereiches

Funktion:

Diese Funktion transportiert den Kursor auf die letzte Zeile des angezeigten Textbereiches.

Format **[BOTTOM EDGE]**

Taste <CTRL>+<PgDn>

Anwendung:

Mit dieser Funktion kann der Kursor schnell auf die letzte angezeigte Zeile des angezeigten Textbereiches positioniert werden. Die Spaltenposition in der Zeile wird beibehalten.

Sollte sich der Kursor auf der Befehlszeile befinden, wird an den Anfang der letzten angezeigten Zeile positioniert.

Siehe auch die Funktion [TOP EDGE]!

2.4.7. CENTER LINE - Zentrieren Zeile

Funktion:

Die Zeile, die den Kursor enthaelt, wird in die Mitte des angezeigten Textfensters verschoben.

Format [CENTER LINE]

Taste keine Zuordnung

Anwendung:

Mittels dieser Funktion kann die durch den Kursor markierte Zeile in die Mitte des angezeigten Textfensters verschoben werden. Es wird dabei keine Ruecksicht auf die Position des Kursors genommen.

Die Zeile, die den Kursor enthaelt wird immer auf die 10. Zeile eingestellt.

Befindet sich der Kursor auf der Befehlszeile, wird diese Funktion ignoriert.

2.4.8. COMMAND TOGGLE - Umschalten Editiermodus-Kommandomodus

Funktion:

Diese Funktion schaltet vom Editiermodus in den Kommandoeingabemodus um und wieder zurueck.

Format [COMMAND TOGGLE]

Taste <ESC>

Anwendung:

Diese Funktion wird verwendet, wenn der Eingabemodus gewechselt werden soll. Soll ein Kommando eingeben werden und der Editiermodus ist eingestellt, muss vorher mit dieser Funktion in den Kommandoeingabemodus umgeschaltet werden. Der Kursor wird aus der Dateizeile in die Befehlszeile transportiert. Erst jetzt kann das Kommando eingegeben werden. Soll in der Datei weitergearbeitet werden, ist erneut mit dieser Funktion umzuschalten.

Beim Uebergang in den Kommandoeingabemodus wird die aktuelle Kursorposition gerettet. Beim Zurueckschalten steht der Kursor wieder an der alten Stelle.

Siehe auch die Funktionen [CORSOR COMMAND] und [CURSOR DATA]!

2.4.9. CONFIRM CHANGE - Bestaetigung des Austausches bei Suchen mit Austauschen

Funktion:

Ein Zeichen bzw. eine Zeichenfolge wird durch ein anderes Zeichen bzw. eine andere Zeichenfolge ersetzt.

Format [CONFIRM CHANGE]

Taste <SHIFT>+<F5>

Anwendung:

Diese Funktion ist nur anwendbar, wenn vorher das CHANGE-Kommando ausgefuehrt und in diesem Kommando nicht der Globalparameter angegeben wurde. Durch Druecken dieser Tasten wird der Austausch vollzogen.

Wird diese Funktion an anderer Stelle aufgerufen, erscheint in der Hinweiszeile die Fehlermeldung:

Kein Austausch

Siehe auch das CHANGE-Kommando!

2.4.10. COPY MARK - Kopieren markierter Bereich

Funktion:

Diese Funktion kopiert einen markierten Bereich an die aktuelle Kursorposition. Das Original bleibt erhalten.

Format [COPY MARK]

Taste <ALT>+<Z>

Anwendung:

Fuer einen markierten Zeilenbereich:

Der markierte Zeilenbereich wird als Kopie an der Kursorposition eingefuegt.

Fuer einen markierten Block:

Die Kopie des markierten Blockes wird an der aktuellen Kursorposition eingefuegt. Der Kursor markiert dabei die linke Spalte des Blockes. Der Text, der links vom Kursor steht, bleibt links stehen. Der Text, der ab der Kursorposition steht, steht dann rechts neben dem Block. Die Zeilenanzahl wird dabei nicht geaendert.

Fuer einen markierten Zeichenbereich:

Es wird die Kopie des markierten Bereiches ab der Kursorposition in der aktuellen Zeile eingefuegt. Die Zeilenanzahl erhoeht sich entsprechend der eingefuegten Zeilen.

Diese Funktion kann fuer die Vorbereitung einer nur zum Teil benoetigten Datei zum Druck verwendet werden.

Dabei ist folgendes auszufuehren:

1. Markieren des zu druckenden Bereiches als Block
2. Eroeffnen einer neuen Datei
3. Einfuegen des markierten Blockes in diese Datei
4. Ausdrucken dieser Datei

Wird die Kopie eines Blockes einer Datei in einer anderen Datei benoetigt, ist folgendermassen zu verfahren:

1. Der Text ist in der Orginaldatei zu markieren (mit Zeilenmarkierung)
2. Auswahl einer anderen Datei.
3. Positionieren des Kursors an die gewuenschte Stelle der aktuellen Datei.
4. Einfuegen des Blockes mittels Blockkopieren in die aktuelle Datei (Tasten <ALT>+<Z>).
5. Weiterarbeit, bzw. Wiederholung der Punkte 2-4.

Die [COPY MARK]-Funktion darf nicht auf den markierten Bereich selbst oder in einen markierten Bereich angewendet werden. In der Hinweiszeile erscheint die Fehlermeldung:

Fehler Quelle und Ziel

Siehe auch die Funktionen [MARK BLOCK], [MARK CHAR], [MOVE MARK] und [MARK LINE]!

2.4.11. CURSOR COMMAND - Kursor auf Befehlszeile

Funktion:

Diese Funktion transportiert den Kursor auf die Befehlszeile.

Format [CURSOR COMMAND]

Taste keine Zuordnung

Anwendung:

Sollte sich der Kursor bereits in der Befehlszeile befinden und diese Funktion wird aufgerufen, erfolgt keine Wirkung. Steht der Kursor beim Aufrufen der Funktion im angezeigten Textbereich, wird der Kursor in die Befehlszeile transportiert und seine vorherige Position gemerkt. Diese Funktion ist kein Schalter. Sie ist vorrangig fuer das Zuweisen von Funktionen zu den einzelnen

Tasten vorgesehen. Diese Funktion stellt das Gegenteil zur Funktion [CURSOR DATA] dar.

Siehe auch die Funktionen [COMMAND TOGGLE] und [CURSOR DATA]!

2.4.12. CURSOR DATA - Kursor in den Datenbereich

Funktion:

Diese Funktion transportiert den Kursor von der Befehlszeile zurueck an die gemerkte Kursorposition in den Datenbereich.

Format **[CURSOR DATA]**

Taste Keine Zuordnung

Anwendung:

Sollte sich der Kursor bereits im Datenbereich befinden, zeigt diese Funktion keine Wirkung. Befindet er sich jedoch auf der Befehlszeile, dann transportiert diese Funktion den Kursor an die gemerkte Stelle in die Datei zurueck. Diese Funktion wird vorrangig fuer Funktionszuweisungen zu einzelnen Tasten verwendet. Sie stellt das Gegenteil zur Funktion [CURSOR COMMAND] dar.

Siehe auch die Funktionen [COMMAND TOGGLE] und [CURSOR COMMAND]!

2.4.13. DELETE CHAR - Loeschen Zeichen

Funktion:

Das Zeichen auf der Kursorposition wird geloescht und der nachfolgende Text der Zeile ein Zeichen nach links verschoben.

Format **[DELETE CHAR]**

Taste <DEL>

Anwendung:

Diese Funktion wird zur Korrektur des Textes verwendet. Fuer die Sofortkorrektur eignet sich aber die [RUBOUT]-Funktion besser, da diese das Zeichen links vom Kursor loescht.

Siehe auch die Funktion [RUBOUT]!

2.4.14. DELETE LINE - Loeschen Zeile

Funktion:

Die Zeile, die den Kursor enthaelt wird geloescht und der nach-

folgende Text um jeweils eine Zeile nach oben verschoben.

Format **[DELETE LINE]**

Taste <CTRL>+<BACKSPACE>

Anwendung:

Diese Funktion wird zum Loeschen von Zeilen verwendet. Der nachfolgende Text wird entsprechend der geloeschten Zeilenanzahl nach oben verschoben. Der Kursor verbleibt beim Loeschen an seiner Position. Befindet sich der Kursor auf der Kommandozeile, wirkt diese Funktion wie <BACKSPACE>.

Siehe auch die Funktion [INSERT LINE] und das Kommando ERASE!

2.4.15. DELETE MARK - Loeschen eines markierten Bereiches

Funktion:

Diese Funktion loescht einen markierten Bereich einschliesslich der Markierungen in der Datei.

Format **[DELETE MARK]**

Taste <ALT>+<D>

Anwendung:

Fuer Blockmarkierungen:

Das markierte Rechteck wird geloescht und der rechts stehende Text nach links verschoben. Die Zeilenlaenge aendert sich.

Fuer Zeilenmarkierungen:

Der mit Zeilenmarkierungen versehene Text wird geloescht. Der nachfolgende Text wird um die Anzahl der geloeschten Zeilen nach oben verschoben.

Fuer Zeichenmarkierungen:

Ein durch Zeichenmarkierungen eingeschlossener Text wird entsprechend den Funktionen [ERASE END LINE] und [ERASE BEGIN LINE] geloescht. Der nachfolgende Text wird nachgerueckt. Der Kursor bleibt stationaer.

Siehe auch die Funktionen [UNMARK], [MARK BLOCK], [MARK CHAR] und [MARK LINE]!

2.4.16. DOWN = Kursor eine Zeile tiefer

Funktion:

Der Kursor wird eine Zeile tiefer gesetzt (in Richtung Dateiende).

Format **[DOWN]**

Taste <CURSOR DOWN>

Anwendung:

Diese Funktion positioniert den Kursor eine Zeile tiefer, wobei die Spaltenposition beibehalten wird. Steht der Kursor beim Funktionsaufruf auf der letzten angezeigten Zeile, wird der gesamte angezeigte Textbereich um eine Zeile nach oben gerollt. Der Kursor bleibt in der letzten Zeile stehen.

Befindet sich der Kursor beim Aufruf der Funktion auf der Befehlszeile, wird der gesamte angezeigte Textbereich um eine Zeile nach oben gerollt und der Kursor auf die erste Position der letzten Textzeile gesetzt. Steht der Kursor auf der letzten Zeile der Datei, erfolgt keine Aenderung der Position.

Siehe auch die Funktionen [UP] und [DOWN4]!

2.4.17. DOWN4 = Kursor um 4 Zeilen tiefer

Funktion:

Der Kursor wird 4 Zeilen nach unten (in Richtung Dateiende) transportiert.

Format **[DOWN4]**

Taste Keine Zuordnung

Anwendung:

Mit dieser Funktion kann der Kursor jeweils in Viererschritten in Richtung Dateiende positioniert werden.

Beim Ueberschreiten des angezeigten Textbereiches wird der angezeigte Text um die entsprechende Zeilenzahl nach oben gerollt.

Befindet sich der Kursor beim Aufrufen dieser Funktion auf der Befehlszeile, wird der Textausschnitt nach oben gerollt und der Kursor an den Anfang der letzten angezeigten Zeile gestellt.

Siehe auch die Funktionen [UP4] und [DOWN]!

2.4.18. END LINE - Kursor an das Ende der Zeile

Funktion:

Diese Funktion transportiert den Kursor an das Ende der Zeile (nach dem letzten Zeichen auf der Zeile).

Format [END LINE]

Taste <END>

Anwendung:

Ist das Ende der Zeile nicht im sichtbaren Textbereich, wird der Text horizontal nach links gerollt, so dass der Kursor in der Mitte des Bildschirmes steht. Steht der Kursor bereits am Ende der Zeile, ist die Funktion wirkungslos. Befindet sich der Kursor in einer Leerzeile, wird er in die Spalte 1 positioniert.

Siehe auch die Funktion [BEGIN LINE]!

2.4.19. END MARK - Kursor an das Ende eines markierten Bereiches

Funktion:

Diese Funktion setzt den Kursor auf das letzte Zeichen eines markierten Bereiches.

Format [END MARK]

Taste <ALT>+<E>

Anwendung:

Der Kursor wird bei Block- und Zeichenmarkierungen exakt unter das letzte markierte Zeichen transportiert.

Bei Zeilenmarkierungen wird der Kursor nur vertikal in der Spalte in die letzte markierte Zeile gesetzt. Diese Funktion schaltet auch von einer aktiven Datei in eine andere um, wenn sich die Markierung nicht in der aktuellen Datei befindet.

Siehe auch die Funktion [BEGIN MARK]!

2.4.20. ERASE BEGIN LINE - Loeschen bis Zeilenanfang

Funktion:

Alle Zeichen ab Zeilenanfang bis zur Kursorposition werden geloescht.

Format **[ERASE BEGIN LINE]**

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird verwendet, wenn ein groesserer Teil einer Zeile zusammenhaengend ab dem Beginn der Zeile geloescht werden soll. Der Kursor bleibt beim Loeschvorgang an seiner Position stehen. Der Rest der Zeile wird an den Zeilenanfang verschoben.

Siehe auch die Funktion [ERASE END LINE]!

2.4.21. ERASE END LINE - Loeschen bis Zeilenende

Funktion:

Alle Zeichen ab der Kursorposition bis zum Ende der aktuellen Zeile werden geloescht.

Format **[ERASE END LINE]**

Taste <F6>

Anwendung:

Diese Funktion wird verwendet, um einen zusammenhaengenden Textbereich bis zum Ende einer Zeile zu loeschen. Der Kursor bleibt an seiner Position stehen. Er markiert das Zeilenende.

Siehe auch die Funktion [ERASE BEGIN LINE]!

2.4.22. ESCAPE - Eingabe von ASCII-Zeichen in den Text

Funktion:

Diese Funktion ermoeglicht die Eingabe von Zeichen aus dem gesamten ASCII-Zeichenvorrat in die Datei.

Format **[ESCAPE]**

Taste <ALT>+<X>

Anwendung:

Sonderzeichen, die nicht auf der Tastatur vorhanden sind, koennen eingegeben werden.

Es wird die Tastenkombination <ALT>+<X> bedient. Anschliessend erfolgt die Eingabe des Sonderzeichens. Das Sonderzeichen wird in seiner dezimalen Codierung zusammen mit der <ALT>-Taste auf dem Zehnerblock der Tastatur eingegeben.

Beispiele:

1. Es soll ein Seitenvorschub (Codierung Ø12) in den Text eingefuegt werden. Dazu sind folgende Schritte notwendig:

- <ALT>+<X>-Taste bedienen
- <ALT>+<Ø>, <ALT>+<1> und <ALT>+<2> eingeben

Mit dem Loslassen der <ALT>-Taste wird dann das entsprechende Zeichen eingestellt.

2. Der Tastenkombination <ALT>+<4> wird das Wurzelzeichen zugeordnet (dezimale Codierung 251). Es ist dann nur noch diese Tastenkombination zu betaetigen, um dieses Sonderzeichen zu erhalten. Folgende Zuweisung in der Datei BE.PRO oder in einem Makro ist vorzunehmen:

```
def a-4 = 'V¯'
```

Siehe dazu auch die Liste der ASCII-Zeichenzuordnung und deren dezimale Codierung!

Die Eingabe der Zeichencodierung muss auf dem Zehnerblock der Tastatur erfolgen.

Siehe auch die Kommandos DEFINE und MACRO!

2.4.23. EXECUTE - Ausfuehrung eines Kommandos

Funktion:

Ein in der Kommandozeile stehendes Kommando wird wiederholt abgearbeitet.

Format **[EXECUTE]**

Taste <CTRL>+<ENTER>

Anwendung:

Solange das Kommando, z.B. CHANGE oder LOCATE, in der Befehlszeile steht, kann es wiederholt werden.

Darueber hinaus wird diese Funktion in Funktionszuweisungen zu den Tasten benoetigt.

Siehe auch das Kommando DEFINE!

2.4.24. FILL MARK - Fuellen eines markierten Bereiches

Funktion:

Diese Funktion fuellt einen markierten Bereich mit einem eingegebenen Zeichen.

Format **[FILL MARK]**

Taste <ALT>+<F>

Anwendung:

Wenn ein markierter Bereich mit einem bestimmten Zeichen gefuellt werden soll, ist diese Funktion zu benutzen. Nach Bedienen dieser Tasten erwartet das Programm die Eingabe des zum Fuellen zu verwendenden Zeichens. Ist das Fuellzeichen ein Sonderzeichen aus dem ASCII-Zeichenvorrat, muss dieses mittels der [ESCAPE]-Funktion eingeben werden.

Siehe dazu auch die Funktionen [ESCAPE], [MARK BLOCK] [MARK CHAR] und [MARK LINE] !

2.4.25. FIND BLANK LINE - Suche folgende Leerzeile

Funktion:

In der Datei ab der Kursorposition vorwaerts wird die naechste Leerzeile gesucht.

Format **[FIND BLANK LINE]**

Taste Keine Zuordnung

Anwendung:

Diese Funktion ist zu verwenden, wenn der Kursor an die naechste Leerzeile positioniert werden soll. Steht der Kursor bereits auf einer Leerzeile, erfolgt keine Positionierung. Wird eine Leerzeile gefunden, wird der Kursor auf die erste Position gestellt; ansonsten erfolgt die Fehlermeldung

Nicht gefunden

auf der Hinweiszeile. Es werden nur echte Leerzeilen gefunden, d.h. es darf sich kein weiteres Zeichen auf der Zeile befinden.

Eingerueckte Zeilen oder mit der Funktion [REFLOW] neu formatierte Zeilen werden nicht als leer betrachtet.

Siehe auch die Funktion [REFLOW]!

2.4.26. FIRST NONBLANK - Suche erstes Textzeichen

Funktion:

Der Kursor wird unter das erste Textzeichen in der aktuellen Zeile, das kein Leerzeichen ist, transportiert.

Format [FIRST NONBLANK]

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird benoetigt, wenn der Kursor an das erste Zeichen einer eingerueckten Zeile transportiert werden soll. Sie eignet sich besonders fuer die Korrektur von Programmquelltexten.

2.4.27. INDENT - Kursor an Beginn der Absatzeinrueckung

Funktion:

Der Kursor wird an den eingerueckten linken Rand eines Absatzes gesetzt.

Format [INDENT]

Taste Keine Zuordnung

Anwendung:

Diese Funktion positioniert den Kursor an die eingerueckte Absatzposition, wenn sich der Kursor zuvor auf einer Leerzeile oder am Dateianfang befunden hat. Ansonsten wird er auf die erste Position in der Zeile transportiert.

Wurde der linke Rand einer Absatzeinrueckung nicht eingestellt (mit dem SET MARGINS Kommando), wird der Kursor ebenfalls auf die erste Position transportiert.

Diese Funktion wird vorrangig fuer die Funktionszuweisung zu den einzelnen Tasten verwendet.

Beispiel:

1. Der <ENTER>-Taste sollen folgende Funktionen zugewiesen wer-

den:

[begin line], [down] und [indent]

Diese Zuweisung transportiert den Kursor an den Beginn der folgenden Zeile. Die Zuordnung eignet sich besonders zur Programmerfassung.

2. Der <ENTER>-Taste sollen die Funktionen

[insert line] und [indent]

zugewiesen werden. Mit dem ersten Druecken der <ENTER>-Taste wird der Kursor an den Anfang der naechsten Zeile gebracht und mit dem zweiten zur Position des linken Absatzrandes.

Siehe auch die Funktionen [BEGIN LINE], [DOWN], [INSERT LINE] und das Kommando SET MARGINS!

2.4.28. INSERT LINE - Einfuegen einer Leerzeile

Funktion:

Eine Leerzeile wird nach der aktuellen Zeile eingefuegt.

Format **[INSERT LINE]**

Taste <F9>

Anwendung:

Diese Funktion fuegt eine Leerzeile nach der aktuellen Zeile ein und transportiert den Kursor an den linken Rand der eingefuegten Zeile.

Sollen vor die erste Zeile in der Datei Leerzeilen eingefuegt werden, muss dazu die Funktion [SPLIT] verwendet werden.

Die Funktion [INSERT LINE] zeigt keine Wirkung, wenn der Kursor auf der Befehlszeile steht.

Siehe auch die Funktionen [DELETE LINE] und [SPLIT]!

2.4.29. INSERT MODE - Einfuegemodus einschalten

Funktion:

Der Einfuegemodus wird eingeschaltet.

Format **[INSERT MODE]**

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird vorrangig in Makros verwendet. Sie bewirkt das unbedingte Umschalten auf den Einfuegemodus, d.h. alle eingegebenen Zeichen werden ab der Kursorposition eingefuegt. Der Rest der Zeile wird nach rechts verschoben. In der Statuszeile wird das Wort "Einfg." angezeigt.

Siehe auch die Funktionen [INSERT TOGGLE] und [REPLACE MODE]!

2.4.30. INSERT TOGGLE = Ueberschreibemodus einschalten

Funktion:

Diese Funktion schaltet vom Einfuegemodus in den Ueberschreibemodus und zurueck.

Format **[INSERT TOGGLE]**

Taste <INS>

Anwendung:

Wenn im Ueberschreibemodus gearbeitet wird (in der Statuszeile steht das Wort "Uebers.") und es wird diese Funktion aufgerufen, wird in den Einfuegemodus umgeschaltet. In der Statuszeile steht dann das Wort "Einfg.". Wird diese Funktion ein weiteres Mal aufgerufen, wird wieder zurueck in den Ueberschreibemodus geschaltet. An Hand der Mitteilung in der Statuszeile kann ueberprueft werden, welche Eingabeart aktuell eingestellt ist.

Im Einfuegemodus werden alle von der Tastatur eingegebenen Zeichen vor der Kursorposition eingefuegt und der Rest der Zeile wird nach rechts verschoben. In der Eingabeart Ueberschreiben wird das durch den Kursor markierte Zeichen von einem von der Tastatur eingegebenen Zeichen ueberschrieben.

Siehe auch die Funktionen [INSERT MODE] und [REPLACE MODE]!

2.4.31. JOIN = Zwei Zeilen miteinander verbinden

Funktion:

Diese Funktion verbindet die aktuelle Zeile mit der nachfolgenden Zeile, so dass eine Zeile entsteht.

Format **[JOIN]**

Taste <ALT>+<J>

Anwendung:

Sollen zwei benachbarte Zeilen zu einer Zeile verbunden werden, ist diese Funktion zu verwenden. Der Kursor wird in die erste Zeile positioniert und anschliessend diese Funktion aufgerufen. Im Ergebnis entsteht dann eine Zeile, die dem Inhalt von beiden entspricht.

Ergibt sich nach dem Verbinden eine Zeilenlaenge von mehr als 255 Zeichen, gehen die Zeichen ab der Stelle 256 verloren. Es erscheint in der Hinweiszeile die Fehlermeldung:

Zeilenueberlauf

Aus der intern mitgefuehrten Datei ".UNNAMED" kann der alte Zustand der Datei wieder zurueckgeholt werden.

Diese Funktion ist nicht anzuwenden, wenn sich der Kursor auf der Befehlszeile befindet.

Siehe auch die Funktion [SPLIT]!

2.4.32. LEFT - Kursor ein Zeichen nach links

Funktion:

Der Kursor wird ein Zeichen nach links bewegt.

Format **[LEFT]**

Taste <KURSOR LINKS>

Anwendung:

Mit dieser Funktion kann der Kursor jeweils um ein Zeichen nach links bewegt werden. Wenn sich der Kursor am linken Rand bzw. in der ersten Spalte befindet, erfolgt keine Positionierung.

Siehe auch die Funktionen [RIGHT], [LEFT8] und [LEFT40]!

2.4.33. LEFT8 - Kursor 8 Zeichen nach links

Funktion:

Der Kursor wird jeweils um 8 Zeichen nach links bewegt.

Format **[LEFT8]**

Taste Keine Zuordnung

Anwendung:

Diese Funktion wirkt wie die Funktion [left], nur dass der Kursor mit jedem Aufruf um 8 Zeichen nach links verschoben wird.

Ist der linke Rand bzw. die Spalte 1 erreicht, erfolgt keine Verschiebung.

Ueberschreitet der Kursor den linken Rand des angezeigten Textes, weil der linke Rand der Zeile noch nicht erreicht ist, wird der gesamte angezeigte Textbereich nach rechts gerollt, so dass der Kursor in der Mitte des Bildschirmes auf der Zeile steht.

Siehe auch die Funktionen [RIGHT8] und [LEFT]!

2.4.34. LEFT40 - Kursor 40 Zeichen nach links

Funktion:

Der Kursor wird um jeweils 40 Zeichen nach links bewegt.

Format [LEFT40]

Taste <CTRL>+<KURSOR LINKS>

Anwendung:

Diese Funktion wird verwendet, um den Kursor ueber einen Bereich von 40 Zeichen nach links zu positionieren . Der Kursortransport endet in jedem Falle in der Position 1 bzw. am linken Rand der Zeile. Ansonsten gelten die Erlaeuterungen zur Funktion [LEFT8].

Siehe auch die Funktionen [RIGHT40], [LEFT8] und [LEFT]!

2.4.35. LEFT_EDGE - Kursor an das 1. Zeichen der Zeile

Funktion:

Der Kursor wird unter das am weitesten links stehende Zeichen der aktuellen Zeile im angezeigten Textbereich positioniert.

Format [LEFT EDGE]

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird verwendet, um den Kursor unter das erste Zeichen in der aktuellen Zeile des angezeigten Textbereiches zu positionieren, unabhaengig davon, wo er sich gerade befindet.

Diese Funktion wirkt nur in dem angezeigten Textbereich. Es erfogt keine Veraenderung der Anzeige.

Siehe auch die Funktion [RIGHT EDGE]!

2.4.36. LOWERCASE - Wandeln Gross- in Kleinbuchstaben

Funktion:

Diese Funktion wandelt alle Grossbuchstaben in einem markierten Bereich in Kleinbuchstaben.

Format [LOWERCASE]

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird verwendet, um in einem markierten Bereich alle grossgeschrieben Buchstaben in Kleinbuchstaben umzuwandeln. Die Anordnung der Zeichen wird nicht veraendert.

Wird diese Funktion in einem nichtmarkierten Bereich aufgerufen, erscheint in der Hinweiszeile die Fehlermeldung:

Markierung fehlt

Siehe auch die Funktion [UPPERCASE]!

2.4.37. MARK BLOCK - Einstellen Blockmarkierungen

Funktion:

Diese Funktion markiert die Eckpunkte eines Blockes.

Format [MARK BLOCK]

Taste <ALT>+<B>

Anwendung:

Es ist die Spalte der linken oberen Ecke anzugeben und die Spalte der rechten unteren Ecke. Ein Block kann sich ueber mehrere Zeilen erstrecken. Der markierte Block wird hervorgehoben angezeigt.

Es ist nur eine Markierung in den aktiven Dateien zugelassen. Wird die Funktion aufgerufen und es besteht schon eine Markierung, erscheint in der Hinweiszeile die Fehlermeldung:

Block ist bereits markiert

Siehe auch die Funktionen [UNMARK], [DELETE MARK], [COPY MARK] [MOVE MARK] und [OVERLAY BLOCK]!

2.4.38. MARK CHAR - Einstellen Zeichenmarkierungen

Funktion:

Diese Funktion markiert einen Zeichenbereich.

Format [MARK CHAR]

Taste <ALT>+<C>

Anwendung:

Diese Funktion wird zum Markieren eines Zeichenbereiches bzw. eines Zeichens oder eines Wortes verwendet. Der markierte Bereich wird hervorgehoben dargestellt. Die Markierung kann sich ueber mehrere Zeilen erstrecken.

In den aktiven Dateien kann nur eine Markierung eingestellt werden. Existiert bereits eine Markierung und es soll eine weitere vorgenommen werden, erscheint auf der Hinweiszeile die Fehlermeldung:

Block ist bereits markiert

Siehe auch die Funktionen [UNMARK], [DELETE MARK] und [COPY MARK]!

2.4.39. MARK LINE - Einstellen Zeilenmarkierungen

Funktion:

In der aktuellen Datei wird eine Zeile bzw. einen Zeilenbereich markiert.

Format [MARK LINE]

Taste <ALT>+<L>

Anwendung:

Diese Funktion wird zum Markieren von Zeilen verwendet, die durch eine andere Funktion behandelt werden koennen. Der markierte Bereich wird hervorgehoben angezeigt. Die Stellung des Kursors in der Zeile ist beliebig. Es wird immer die gesamte Zeile von der Position 1 bis zur Position 255 angenommen.

Es ist nur ein markierter Bereich in den aktiven Dateien zugelassen. Ist bereits eine Markierung vorhanden, wenn diese Funktion aufgerufen wird, erscheint auf der Hinweiszeile die Fehlermeldung:

Block ist bereits markiert

Siehe auch die Funktionen [UNMARK], [DELETE MARK] und [COPY MARK]!

2.4.40. MOVE MARK - Verschieben eines markierten Bereiches

Funktion:

Ein markierter Bereich wird an die Kursorposition eingefuegt und der Originalbereich geloescht.

Format [MOVE MARK]

Taste <ALT>+<M>

Anwendung:

Diese Funktion wird zum Verschieben von Textbereichen benoetigt. Damit kann eine Umordnung des Textes vorgenommen werden. Die Wirkung ist folgende:

Fuer Zeilenmarkierungen:

Der markierte Textbereich wird vor der aktuellen Kursorposition eingefuegt. Der Rest der Zeile ab der Kursorposition steht im Anschluss an den eingefuegten Bereich. Der Kursor bleibt auf dem Zeichen stehen, das er vor der Verschiebung markiert hat.

Fuer Blockmarkierungen:

Der markierte Block wird an der Kursorposition eingefuegt. Der Kursor markiert dabei die linke obere Ecke des Blockes. Der links von der Kursorposition stehende Text bleibt als Spaltenbereich stehen. Der Rest der Zeile ab Kursorposition steht im Anschluss an den eingefuegten Block (entsprechend der Blockbreite nach rechts verschoben).

Fuer Zeichenmarkierungen:

Der markierte Bereich wird vor der Kursorposition eingefuegt. Der Rest der Zeile ab Kursorposition steht im Anschluss an den eingefuegten Bereich.

Wird diese Funktion aufgerufen und es ist keine Markierung vorhanden, erscheint auf der Hinweiszeile die Fehlermeldung:

Markierung fehlt

Siehe auch die Funktion [COPY BLOCK]!

2.4.41. OVERLAY BLOCK - Ueberlagern durch einen Block

Funktion:

Ein Textbereich wird entsprechend der Groesse des markierten Blockes ueberlagert.

Format [OVERLAY BLOCK]

Taste <ALT>+<O>

Anwendung:

Diese Funktion wird verwendet, um einen Textbereich durch einen markierten Block zu ueberschreiben. Sie ist nur anwendbar, wenn der Bereich mit Blockmarkierungen versehen ist. Es wird das als Block gekennzeichnete Rechteck zur Ueberlagerung verwendet. Die Kursorposition markiert dabei die linke obere Ecke des Rechteckes. Wird diese Funktion aufgerufen und es ist kein mit Blockmarken versehener Bereich in der Datei vorhanden, erscheint auf der Hinweiszeile die Fehlermeldung:

Blockmarkierung fehlt

Siehe auch die Funktionen [COPY MARK] und [MOVE MARK]!

2.4.42. PAGE DOWN - Blaettern eine Seite vorwaerts

Funktion:

Der angezeigte Textbereich wird um eine Bildschirmseite nach oben gerollt.

Format [PAGE DOWN]

Taste <PgDn>

Anwendung:

Diese Funktion wird verwendet, um seitenweise die Datei durchzublaettern. 22 Zeilen werden total im Textfenster angezeigt. Der angezeigte Text wird um 20 Zeilen nach oben verschoben und von unten her 20 neue Zeilen aus der Datei eingelesen. Die Kursorposition in der Zeile wird beim Durchblaettern nicht veraendert.

Siehe auch die Funktion [PAGE UP]!

2.4.43. PAGE UP - Blaettern eine Seite rueckwaerts

Funktion:

Der angezeigte Textbereich wird um ein Bildschirmbild nach unten gerollt.

Format [PAGE UP]

Taste <PgUp>

Anwendung:

Diese Funktion wird verwendet, um die Datei seitenweise zurueckzublaettern. Das Textfenster umfasst 22 Zeilen. 20 Zeilen davon werden nach unten aus dem Textfenster herausgerollt und 20 vorhergehende Zeilen aus der Datei eingelesen und angezeigt. Die Kursorposition in der Zeile wird nicht veraendert.

Siehe auch die Funktion [PAGE DOWN]!

2.4.44. REDRAW - Reorganisation Bildschirm nach Aenderung der Bildschirmparameter

Funktion:

Der Bildschirm wird geloescht und der Inhalt im neuen Format angezeigt.

Format [REDRAW]

Taste <ALT>+<R>

Anwendung:

Diese Funktion wird benoetigt, wenn waehrend der Arbeit die Bildschirmparameter mit dem SET DISPLAY-Kommando geaendert werden. Durch Aufruf dieses Kommandos wird der Bildschirminhalt entsprechend der neu eingestellten Parameter angezeigt.

Siehe auch das Kommando SET DISPLAY!

2.4.45. REFLOW - Neuformatieren eines markierten Bereiches

Funktion:

Diese Funktion reformiert einen mit Zeilenmarkierungen markierten Textbereich.

Format [REFLOW]

Taste <SHIFT>+<F3>

Anwendung:

Sie wird verwendet, um einen mit Zeilenmarkierungen versehenen Textbereich neu zu formatieren. Ueberschreitet ein Wort den rechten eingestellten Rand, wird es an den Anfang der folgenden Zeile uebernommen. Sollte ein Wort nicht in eine Zeile passen, wird das Wort am rechten Rand abgetrennt und auf der Folgezeile fortgesetzt. Es wird keine automatische Rechtsausrichtung durchgefuehrt. Fuer Absatzneuformatierung sind die Tasten <ALT>+<P> zu betaetigen.

Siehe auch das SET MARGINS-Kommando!

2.4.46. REPLACE MODE - Umschalten auf Ueberschreibemodus

Funktion:

Der Ueberschreibemodus wird eingeschaltet.

Format [REPLACE MODE]

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird vorrangig fuer Funktionszuweisungen zu den Tasten verwendet. Der Ueberschreibemodus wird eingestellt und in der Statuszeile das Wort "Uebers." angezeigt.

Siehe auch die Funktionen [INSERT TOGGLE] und [INSERT MODE]!

2.4.47. RIGHT - Kursor 1 Zeichen nach rechts

Funktion:

Der Kursor wird ein Zeichen nach rechts bewegt.

Format [RIGHT]

Taste <KURSOR RECHTS>

Anwendung:

Diese Funktion wird verwendet, um den Kursor um ein Zeichen nach rechts zu verschieben. Ueberschreitet der Kursor den rechten Rand des angezeigten Textbereiches, wird der Text horizontal nach links gerollt, so dass der Kursor in der Mitte des Bildschirmes steht. Hat der Kursor die Position 255 erreicht, erfolgt keine weitere Verschiebung.

Siehe auch die Funktionen [LEFT], [RIGHT40] und [RIGHT8]!

2.4.48. RIGHT8 = Kursor 8 Zeichen nach rechts

Funktion:

Der Kursor wird um 8 Positionen nach rechts bewegt.

Format **[RIGHT8]**

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird verwendet, um den Kursor in 8-er Schritten vorwaerts zu bewegen. Ueberschreitet der Kursor beim Transport den rechten Rand des angezeigten Textbereiches, wird der Text horizontal nach links gerollt, so dass der Kursor in der Mitte des Bildschirmes steht. Ab der Position 255 erfolgt keine weitere Verschiebung.

Siehe auch die Funktionen [LEFT8], [RIGHT40] und [RIGHT]!

2.4.49. RIGHT40 = Kursor 40 Zeichen nach rechts

Funktion:

Der Kursor wird um 40 Spalten nach rechts bewegt.

Format **[RIGHT40]**

Taste <CTRL>+<KURSOR RECHTS>

Anwendung:

Diese Funktion wird verwendet, um den Kursor in 40-er Spaltenschritten vorwaerts zu positionieren. Dabei wird der Text horizontal nach links gerollt, falls der Kursor den rechten Rand des angezeigten Textbereiches ueberschreitet. Der Kursor steht dann wieder in der Mitte des Bildschirmes. Der Kursortransport endet in jedem Falle an der Position 255 in der Zeile.

Siehe auch die Funktionen [LEFT40], [RIGHT8] und [RIGHT]!

2.4.50. RIGHT EDGE - Kursor zum rechten Rand

Funktion:

Der Kursor wird an den rechten Rand der Zeile des sichtbaren Textbereiches auf dem Bildschirm bewegt.

Format **[RIGHT EDGE]**

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird verwendet, um den Kursor unter das letzte angezeigte Zeichen in der Zeile zu positionieren. Der Kursor kann nicht ueber die Spalte 255 hinaus positioniert werden. Befindet sich der Kursor auf der Befehlszeile, erfolgt keine Positionierung.

Siehe auch die Funktion [LEFT EDGE]!

2.4.51. RUBOUT - Loeschen Zeichen links vom Kursor

Funktion:

Das Zeichen links vom Kursor wird geloescht und der nachfolgende Text der Zeile um ein Zeichen nach links verschoben.

Format **[RUBOUT]**

Taste <BACKSPACE>

Anwendung:

Diese Funktion wird zur Korrektur von Texteingaben verwendet. Mit jeder Tastenbetaetigung wird ein Zeichen, das links vom Kursor steht, geloescht. Der Rest der Zeile wird entsprechend nach links verschoben. Befindet sich der Kursor in der Spalte 1, wird das Zeichen, unter dem der Kursor steht, geloescht und der Rest der Zeile ebenfalls um eine Position nach links verschoben.

Siehe auch die Funktion [DELETE CHAR]!

2.4.52. SHIFT LEFT - Verschieben ein Zeichen nach links

Funktion:

Ein markierter Textbereich wird um eine Position nach links verschoben.

Format [SHIFT LEFT]

Taste <SHIFT>+<F7>

Anwendung:

Diese Funktion wird zur Linksverschiebung des Textes verwendet. Beim Verschieben wird das links vor der Markierung stehende Zeichen bzw. das erste Zeichen eines markierten Bereiches, falls der Bereich ab Position 1 beginnt, geloescht. Fuer grosse markierte Bereiche benoetigt diese Funktion einen laengeren Zeitraum zur Ausfuehrung.

Siehe auch die Funktion [SHIFT RIGHT]!

2.4.53. SHIFT RIGHT - Verschieben ein Zeichen nach rechts

Funktion:

Ein markierter Textbereich wird um ein Zeichen nach rechts verschoben.

Format [SHIFT RIGHT]

Taste <SHIFT>+<F8>

Anwendung:

Diese Funktion wird zur Rechtsverschiebung von Texten verwendet. Bei der Verschiebung wird ein Leerzeichen vor die erste markierte Stelle eingefuegt. Fuer grosse markierte Bereiche wird ein laengerer Zeitraum zur Ausfuehrung benoetigt.

Siehe auch die Funktion [SHIFT LEFT]!

2.4.54. SPLIT - Aufteilen einer Zeile in zwei Zeilen

Funktion:

Die aktuelle Zeile wird an der Kursorpositon in zwei Zeilen aufgeteilt.

Format [SPLIT]

Taste <ALT>+<S>

Anwendung:

Diese Funktion wird benoetigt, um eine Zeile in zwei Zeilen aufzuteilen. Der Text rechts vom Kursor wird an den Anfang der nachfolgenden Zeile verschoben (eingefuegt). Diese Funktion hat keine Wirkung, wenn sich der Kursor in der Befehlszeile befindet. Bei Ausfuehrung dieser Funktion wird der Kursor nicht veraendert.

Siehe auch die Funktion [JOIN]!

2.4.55. TAB - Kursor zur naechsten Tabulatorposition

Funktion:

Der Kursor wird auf die naechste rechts folgende Tabulatorposition transportiert.

Format [TAB]

Taste <TAB>

Anwendung:

Diese Funktion wird verwendet, um einen spaltengerechten Aufbau des Textes zu erreichen. Standardmaessig sind die Tabulatorpositionen aller 8 Positionen (9, 17, 25...) eingestellt. Maximal koennen 20 Tabulatorpositionen gesetzt werden. Die Einstellung erfolgt mit dem SET TABS-Kommando. Die aktuell eingestellten Tabulatoren koennen mit dem Aufruf des QUESTION MARK -Kommandos abgerufen werden.

Ueberschreitet der Kursor bei einer Tabulation den rechten Rand des angezeigten Textbereiches, wird der angezeigte Textbereich horizontal nach links gerollt, so dass sich der Kursor in der Mitte des Bildschirmes befindet.

Siehe auch die Funktionen [BACKTAB] und [TAB WORD] sowie die Kommandos SET TABS und QUESTION MARK!

2.4.56. TAB WORD - Kursor zum naechsten Wort

Funktion:

Der Kursor wird nach rechts unter das erste Zeichen des folgenden Wortes auf der Zeile gesetzt.

Format [TAB WORD]

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird verwendet, um den Kursor wortweise in der Zeile nach rechts zu positionieren. Ein Wort wird definiert durch eine Gruppe von Zeichen, die durch ein bzw. mehrere Leerzeichen voneinander getrennt sind. Ist das Ende der Zeile erreicht, verbleibt der Kursor an seiner Position.

Verlaesst der Kursor den sichtbaren Bereich des angezeigten Textes, erfolgt ein horizontales Rollen des Textes um 20 bzw. 40 Zeichen nach links, je nach eingestellter Anzeigebreite des Bildschirmes.

Siehe auch die Funktionen [BACKTAB WORD] und [TAB]!

2.4.57. TOP = Kursor in erste Zeile der Datei

Funktion:

Der Kursor wird in die erste Zeile der Datei gesetzt.

Format [TOP]

Taste <CTRL>+<HOME>

Anwendung:

Diese Funktion wird verwendet, um den Kursor an den Anfang der Datei zu positionieren. Der Kursor wird dann in die erste Zeile der Datei gestellt. Befindet sich der Kursor bereits am Dateianfang, verbleibt er an seiner Position.

Siehe auch die Funktion [BOTTOM]!

2.4.58. TOP EDGE = Kursor in erste Zeile des angezeigten Textbereiches

Funktion:

Der Kursor wird auf die erste Zeile des angezeigten Textbereiches auf dem Bildschirm positioniert.

Format [TOP EDGE]

Taste <CTRL>+<PgUp>

Anwendung:

Diese Funktion wird verwendet, um den Kursor in die erste Zeile des angezeigten Textbereiches zu transportieren. Der Kursor veraendert dabei die Spaltenposition nicht. Es erfolgt keine

Aenderung des angezeigten Textes.

Siehe auch die Funktion [BOTTOM EDGE]!

2.4.59. UNDO - Rueckgaengigmachen einer Veraenderung

Funktion:

Der Zustand einer Zeile wird so wie er vor einer Veraenderung der Zeile gewesen ist, wieder hergestellt.

Format [UNDO]

Taste <SHIFT>+<F4>

Anwendung:

Diese Funktion wird zum Stornieren einer Veraenderung in der Zeile verwendet. Es wird der Zustand vor der letzten Veraenderung wieder angezeigt. Diese Funktion ist **nicht** anwendbar, wenn der Kursor bereits die Zeile verlassen hat oder die [JOIN]-Funktion aufgerufen wurde. Das Wiederherstellen der Zeile wird aus der intern mitgefuehrten Datei ".UNNAMED" vorgenommen.

2.4.60. UNMARK - Loeschen von Markierungen

Funktion:

Die eingestellten Markierungen werden geloescht.

Format [UNMARK]

Taste <ALT>+<U>

Anwendung:

Diese Funktion wird zum Loeschen aller Markierungen in den aktiven Dateien verwendet. Es erfolgt keine Aenderung des Textes. Befindet sich der markierte Bereich auf dem Bildschirm, wird dieser Bereich wieder normal angezeigt.

Siehe auch die Funktionen [MARK BLOCK], [MARK CHAR] und [MARK LINE]!

2.4.61. UP - Kursor 1 Zeile nach oben

Funktion:

Der Kursor wird eine Zeile nach oben gesetzt.

Format [UP]

Taste <KURSOR UP>

Anwendung:

Diese Funktion wird benoetigt, um den Kursor um eine Zeile nach oben zu transportieren. Befindet sich der Kursor in der ersten Zeile der Datei, erfolgt keine Positionierung mehr. Befindet sich der Kursor in der ersten Zeile des angezeigten Textbereiches und es ist nicht der Dateianfang, wird der Textbereich um eine Zeile nach unten gerollt. Der Kursor bleibt dabei stationaer.

Befindet sich der Kursor beim Aufrufen dieser Funktion in der Befehlszeile, wird der Kursor auf die letzte Zeile des angezeigten Textbereiches positioniert.

Siehe auch die Funktionen [DOWN] und [UP4]!

2.4.62. UPPERCASE - Wandeln Klein- in Grossbuchstaben

Funktion:

Alle Kleinbuchstaben eines markierten Bereiches werden in Grossbuchstaben gewandelt.

Format [UPPERCASE]

Taste Keine Zuordnung

Anwendung:

Diese Funktion wird verwendet, um einen zusammenhaengenden Bereich komplett in Grossbuchstaben zu wandeln. Dieser Bereich ist in Markierungen einzuschliessen, bevor diese Funktion angewendet werden kann.

Siehe auch die Funktion [LOWERCASE]!

2.4.63. UP4 = Kursor 4 Zeilen nach oben

Funktion:

Diese Funktion setzt den Kursor jeweils um 4 Zeilen nach oben (in Richtung Dateianfang).

Format [UP4]

Taste Keine Zuordnung

Anwendung:

Mit dieser Funktion wird der Kursor in 4-er Zeilenschritten in Richtung Dateianfang verschoben. Die Verschiebung endet am Dateianfang. Verlaesst der Kursor durch die Verschiebung den sichtbaren Textbereich, wird der angezeigte Text um die entsprechende Anzahl von Zeilen nach unten gerollt. Der Kursor bleibt dabei in der ersten angezeigten Textzeile auf dem Bildschirm. Die Verschiebung des Kursors erfolgt in der Spalte vertikal

Siehe auch die Funktionen [DOWN4] und [UP]!

2.5. Standardtastaturzuweisung in BE.PRO

Die folgende Tastaturzuordnung ist fuer die Anwendung bei Programmerfassungen vorgesehen.

Tasten-name	zugeordnete Kommandos und Funktionen	Bedeutung	Taste
UP	[UP]	Kursor 1 Zeile nach oben	< ↑ >
DOWN	[DOWN]	Kursor 1 Zeile nach unten	< ↓ >
LEFT	[LEFT]	Kursor 1 Zeichen nach links	< <- > (*)
RIGHT	[RIGHT]	Kursor 1 Zeichen nach rechts	< -> >
PGUP	[PAGE UP]	Angezeigten Textbereich um 1 Bildschirmseite nach unten rollen	<PAGE UP>
PGDN	[PAGE DOWN]	Angezeigten Textbereich um 1 Bildschirmseite nach oben rollen	<PAGE DOWN>

(*) Taste im Block der Kursorsteuertasten

Tastenname	zugeordnete Kommandos und Funktionen	Bedeutung	Taste
HOME	[BEGIN LINE]	Kursor an den Beginn einer markierten Stelle	< ↑ >
END	[END LINE]	Kursor ans Zeilenende	<**END**>
INS	[INSERT TOGGLE]	Eingabemodus umschalten	<**INS**>
DEL	[DELETE CHAR]	Zeichen, unter dem der Kursor steht, loeschen	<**DEL**>
ENTER	[BEGIN LINE] [DOWN] [INDENT]	Kursor auf den Anfang der neuen Zeile; bei Absatzeinrueckung auf entsprechende Position	< <- | >
BACK-SPACE	[RUBOUT]	1 Zeichen links vom Kursor loeschen	< <- > (**)
ESC	[COMMAND TOGGLE]	Umschalten zwischen Editier- und Kommandoeingabemodus	<**ESC**>
TAB	[TAB]	Tabulatorsprung nach rechts	< |<- ->| >
F1	[CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'E PE.HLP' [EXECUTE]	Help-Menue aufrufen	<**F1**>
F2	[CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'SAVE'	Aktuelle Datei zwischenspeichern	<**F2**>
F3	[CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'FILE'	Aktuelle Datei abspeichern und abschliessen	<**F3**>
F4	[CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'QUIT' [EXECUTE]	Aktuelle Datei verwerfen	<**F4**>
F5	[BEGIN LINE] [ERASE END LINE]	Aktuelle Zeile loeschen	<**F5**>

(**) Oberste Zeile der Alphatastatur, rechts

Tasten-name	zugeordnete Kommandos und Funktionen	Bedeutung	Taste
F6	[ERASE END LINE]	Ab Kursorposition bis Zeilenende loeschen	<F6>
F7	[CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'PRINT	Aktuelle Datei drucken	<F7>
F8	[CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'E' [EXECUTE]	Datei eroeffnen	<F8>
F9	[INSERT LINE]	1 Leerzeile einfuegen	<F9>
F10	[INSERT LINE] [UP] [FIRST NONBLANK] [DOWN]	1 Zeile einfuegen und Kursor an Absatzeinrueckung	<F10>
C-LEFT	[LEFT40]	Kursor 40 Zeichen nach links	<CTRL>+< <- > (*)
C-RIGHT	[RIGHT40]	Kursor 40 Zeichen nach rechts	<CTRL>+< -> >
C-PGUP	[TOP EDGE]	Kursor auf 1. Zeichen des angezeigten Textbereiches	<CTRL>+ <PAGE UP>
C-PGDN	[BOTTOM EDGE]	Kursor auf letzte Zeile des angezeigten Textbereiches	<CTRL>+ <PAGE DOWN>
C-HOME	[TOP]	Kursor in die 1. Zeile der Datei	<CTRL>+< 'T >
C-END	[BOTTOM]	Kursor auf die letzte Zeile der Datei	<CTRL>+<E^>
C-ENTER	[EXECUTE]	Wiederholte Abarbeitung eines in der Befehlszeile stehenden Kommandos	<CTRL>+< <- ! >
C-BACK-SPACE	[DELETE LINE]	Aktuelle Zeile loeschen	<CTRL>+ < <- > (**)
C-F1			<CTRL>+<F1>

(*) Taste im Block der Kursorsteuertasten
(**) Oberste Zeile der Alphatastatur, rechts

Tastenname	zugeordnete Kommandos und Funktionen	Bedeutung	Taste
C-F2			<CTRL>+<F2>
C-F3			<CTRL>+<F3>
C-F4			<CTRL>+<F4>
C-F5			<CTRL>+<F5>
C-F6			<CTRL>+<F6>
C-F7			<CTRL>+<F7>
C-F8			<CTRL>+<F8>
C-F9			<CTRL>+<F9>
C-F10			<CTRL>+<F10>
S-TAB	[BACKTAB]	Tabulatorsprung nach links	< >+< |<- ->| > (***)
S-F1	[PAGE DOWN] [BOTTOM EDGE] [DOWN] [DOWN] [CURSOR COMMAND]	Help-Menue nach unten rollen	< >+<F1> (***)
S-F2	[PAGE UP] [TOP EDGE] [UP] [UP] [CURSOR COMMAND]	Help-Menue nach oben rollen	< >+<F2> (***)
S-F3	[REFLOW]	Mit Zeilenmarkierungen versehenen Text formatieren	< >+<F3> (***)
S-F4	[UNDO]	Veraenderungen in einer Zeile stornieren	< >+<F4> (***)
S-F5	[CONFIRM CHANGE]	Austauschen	< >+<F5> (***)
S-F6			< >+<F6> (***)
S-F7	[SHIFT LEFT]	Markierten Textbereich 1 Position nach links verschieben	< >+<F7> (***)
S-F8	[SHIFT RIGHT]	Markierten Textbereich 1 Position nach rechts verschieben	< >+<F8> (***)

(***) < >-Umschalttaste; Taste ohne Beschriftung

Tasten-name	zugeordnete Kommandos und Funktionen	Bedeutung	Taste
S-F9	[CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'DIR A: [EXECUTE]	Verzeichnis vom Laufwerk A anzeigen	< >+<F9> (***)
S-F10	[CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'DIR B: [EXECUTE]	Verzeichnis vom Laufwerk B anzeigen	< >+<F10> (***)
A-A			<ALT>+<A>
A-B	[MARK BLOCK]	Block markieren	<ALT>+<B>
A-C	[MARK CHAR]	Zeichenbereich markieren	<ALT>+<C>
A-D	[DELETE MARK]	Markierten Bereich loeschen	<ALT>+<D>
A-E	[END MARK]	Kursor auf letztes Zeichen eines markierten Bereichs	<ALT>+<E>
A-F	[FILL MARK]	Markierten Bereich mit angegebenem Zeichen fuellen	<ALT>+<F>
A-G			<ALT>+<G>
A-H			<ALT>+<H>
A-I			<ALT>+<I>
A-J	[JOIN]	Zwei benachbarte Zeilen verbinden	<ALT>+<J>
A-K			<ALT>+<K>
A-L	[MARK LINE]	Zeilenbereich markieren	<ALT>+<L>
A-M	[MOVE MARK]	markierten Bereich verschieben	<ALT>+<M>
A-N			<ALT>+<N>
A-O	[OVERLAY BLOCK]	Textbereich durch markierten Block ueberschreiben	<ALT>+<O>

(***) < >-Umschalttaste; Taste ohne Beschriftung

Tastenname	zugeordnete Kommandos und Funktionen	Bedeutung'	Taste
A-P	[CURSOR DATA] [UNMARK] [MARK LINE] [FIND BLANK LINE] [UP] [MARK LINE] [REFLOW] [END MARK] [DOWN] [DOWN] [UNMARK]	Absatz innerhalb der eingestellten Raender formatieren	<ALT>+<P>
A-Q			<ALT>+<Q>
A-R	[REDRAW]	Text in neuer Form nach Aenderung der Bildschirmparameter anzeigen	<ALT>+<R>
A-S	[SPLIT]	Zeile an der Kursorposition teilen	<ALT>+<S>
A-T			<ALT>+<T>
A-U	[UNMARK]	Markierungen loeschen	<ALT>+<U>
A-V			<ALT>+<V>
A-W			<ALT>+<W>
A-X	[ESCAPE]	Eingeben nicht auf der Tastatur vorhandenes Sonderzeichen	<ALT>+<X>
A-Y	[BEGIN MARK]	Kursor an Beginn eines markierten Bereichs	<ALT>+<Y>
A-Z	[COPY MARK]	Markierten Bereich kopieren	<ALT>+<Z>
A-F1			<ALT>+<F1>
A-F2			<ALT>+<F2>
A-F3			<ALT>+<F3>
A-F4			<ALT>+<F4>
A-F5			<ALT>+<F5>
A-F6			<ALT>+<F6>

Tasten-name	zugeordnete Kommandos und Funktionen	Bedeutung	Taste
A-F7			<ALT>+<F7>
A-F8			<ALT>+<F8>
A-F9			<ALT>+<F9>
A-F10			<ALT>+<F10>
A-0			<ALT>+<0>
A-1			<ALT>+<1>
A-2			<ALT>+<2>
A-3			<ALT>+<3>
A-4			<ALT>+<4>
A-5			<ALT>+<5>
A-6			<ALT>+<6>
A-7			<ALT>+<7>
A-8			<ALT>+<8>
A-9			<ALT>+<9>

Durch Modifizieren der Standardzuweisungen mit folgenden Veraenderungen ist der Editor auch fuer die Textverarbeitung einsetzbar. Es ist dabei die Datei BE.PRO wie folgt zu korrigieren:

```
DEF HOME = [BEGIN LINE] [FIRST NONBLANK]
DEF ENTER = [INSERT LINE] [INDENT]
DEF TAB = [TAB WORD]
DEF F1 = [CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE] 'SAVE'
DEF F2 = [UP4]
DEF F3 = [UNDO]
DEF F4 = [DOWN4]
DEF F5 = [LEFT8]
DEF F6 = [RIGHT8]
DEF F7 = [UNMARK] [MARK CHAR] [TAB WORD] [LEFT] [MARK CHAR]
         [BEGIN MARK] [DELETE MARK]
DEF F8 = [CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE]
         'E' [EXECUTE]
DEF F9 = [ERASE BEGIN LINE]
DEF F10 = [ERASE END LINE]
DEF S-TAB = [BACKTAB WORD]
```

```
DEF A-A = [BEGIN MARK]
DEF A-F = [CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE]
          'FILE'
DEF A-Q = [CURSOR COMMAND] [BEGIN LINE] [ERASE END LINE]
          'Q' [EXECUTE]
DEF A-Y = [COPY MARK]
DEF A-Z = [END MARK]
```

Beachtet

Alle Funktionen und Kommandos, die einer Taste zugewiesen werden sollen, **muessen auf einer Zeile stehen.**

2.6. Fehleranzeigen

Abbruch ? y/n

Diese Ausschrift erfolgt beim Aufruf des QUIT-Kommandos.

Blockmarkierung fehlt

Diese Ausschrift erfolgt beim Aufruf der [OVERLAY BLOCK]-Funktion. Es sind keine Marken gesetzt.

Keine Eroeffnung TMP-Datei

Diese Meldung erfolgt, wenn eine Operation aufgerufen wird, die mehr Speicherplatz beansprucht als aktuell zur Verfuegung steht. Der Editor versucht eine sogenannte TMP-Datei auf dem aktuellen Laufwerk zu eroeffnen, um in diese Datei Teile auszulagern. Diese Eroeffnung war nicht erfolgreich, da das Laufwerk nicht bereit war.

Kein RENAME fuer ".UNNAMED"

Diese Meldung erfolgt, wenn versucht wird, die interne Datei ".UNNAMED" umzubenennen. Interne Dateien koennen nicht umbenannt werden. Wenn der Inhalt dieser Dateien gespeichert werden soll, muss mit der [MARK LINE]-Funktion der Text markiert und mit der [COPY MARK]-Funktion dieser Text in eine andere Datei uebernommen werden. Erst dann kann er wie eine normale Datei behandelt werden.

Kein SAVE fuer interne Datei

Diese Meldung erfolgt, wenn in den Kommandos SAVE bzw. FILE eine der internen Dateien angegeben wurde. Die drei internen Dateien ".DIR", ".UNNAMED" und ".KEYDEFS" koennen nicht mit den Kommandos SAVE oder FILE behandelt werden. Die Dateien ".DIR" und ".KEYDEFS" koennen mit dem Kommando RENAME umbenannt, und danach wie normale Dateien behandelt werden. Das Umbenennen der Dateien kann dann explizit mit dem Kommando NAME oder implizit durch Eingabe eines neuen Dateinamens beim SAVE- oder FILE-Kommando

vorgenommen werden.

Keine Ausfuehrung - Speicher voll

Diese Meldung erfolgt, wenn eine Funktion aufgerufen wurde, diese aber nicht ausgefuehrt werden kann, weil nicht genug Speicherplatz vorhanden ist. Mit dem QUESTION MARK-Kommando kann der noch verfuegbare Speicherbereich angezeigt werden. Es ist Speicherplatz durch Abschluss einer anderen aktiven Datei zu schaffen.

Farb/Grafikadapter nicht installiert

Diese Meldung erfolgt, wenn in der Betriebsart Color gearbeitet werden soll, aber nur ein Schwarz/Weiss-Bildschirm angeschlossen ist. Es ist mit dem SET DISPLAY-Kommando die Betriebsart Schwarz/Weiss einzustellen (mono).

Kommandodatei nicht gefunden

Diese Meldung erfolgt, wenn beim ersten Aufruf des Editors BE die Datei BE.PRO nicht im aktuellen Laufwerk gefunden wurde und wenn die angegebene Datei in einem Makrokommando nicht gefunden wurde.

Kommandodatei fehlerhaft

Diese Meldung erfolgt, wenn in der Kommandodatei BE.PRO eine Makroanweisung angegeben wurde. In der Kommandodatei darf keine Makroanweisung enthalten sein. Wurde ein Syntaxfehler erkannt, erfolgt ebenfalls diese Meldung.

CONFIRM CHANGE

Diese Meldung erfolgt, wenn mit dem CHANGE-Kommando eine Zeichenkette gefunden wurde und die Bestaetigung fuer den Austausch dieser Zeichenkette abverlangt wird. Diese Meldung ist keine Fehlermeldung, sondern ein Hinweis fuer die Weiterarbeit.

Markierungsfehler

Diese Meldung erfolgt, wenn die zweite Markierung (das Ende eines markierten Bereiches) in einer anderen Datei als der die den Beginn des markierten Bereichs enthaltenden eingeben wird. Beide Markierungen muessen in der selben Datei sein. Das trifft fuer alle Markierungsarten zu.

Diskfehler im Verzeichnis

Diese Meldung erfolgt, wenn auf eine nichtinitialisierte Diskette zugegriffen wird. Die Diskette ist erst zu formatieren, bevor mit ihr gearbeitet werden kann.

Diskette voll

Diese Meldung erfolgt, wenn nicht mehr genug Platz auf der

Diskette ist, um die gesamte Datei abzuspeichern. Es ist Platz zu schaffen durch Loeschen von Dateien mittels des ERASE-Kommandos oder die Diskette zu wechseln. Danach ist erneut das SAVE- bzw. FILE-Kommando aufzurufen.

Dateiabschlussfehler

Diese Meldung erfolgt, wenn eine abgespeicherte Datei nicht ordnungsgemaess abgeschlossen werden kann, weil ein Disketten-Fehler vorlag oder das Verzeichnis voll ist. Die Datei sollte auf eine andere Diskette oder ein anderes Laufwerk ausgeben werden.

Eingabefehler in SET DISPLAY

Diese Meldung erfolgt, wenn im SET DISPLAY-Kommando eine unzulaessige Parametereinstellung vorgenommen wurde bzw. ein Syntaxfehler vorlag. Als Parameter sind folgende Typen zugelassen:

"mono", "color 40", "color 80", "b/w 40" oder "b/w 80".

Randeinstellungsfehler

Diese Meldung erfolgt, wenn im SET MARGINS-Kommando eine unzulaessige Parametereinstellung vorgenommen wurde. <n1> bestimmt den linken, <n2> bestimmt den rechten Rand und <n3>, falls benoetigt, die Absatzeinrueckung. Es sind Ziffern von 1 bis 254 (dezimal) zugelassen. <n1> muss kleiner sein als n2.

Fehler in NOTABS-Angabe

Diese Meldung erfolgt, wenn das Wort "NOTABS" nicht korrekt eingegeben wurde, bzw. wenn ein Leerzeichen zwischen einer Laufwerksspezifikation und dem Dateinamen eingegeben wurde. Das Kommando ist erneut einzugeben.

Fehler im Austauschbegriff

Diese Meldung erfolgt, wenn eine Such- bzw. Ersatzzeichenfolge nicht exakt durch die gleichen Begrenzer eingeschlossen ist. In einem CHANGE-Kommando muss der Suchbegriff und der Ersatzbegriff in die gleichen Begrenzungszeichen eingeschlossen sein. Der Ersatzbegriff wurde nicht korrekt eingegeben. Das erste Zeichen, das dem Wort "CHANGE" folgt, wird als Begrenzer der nachfolgenden Zeichenfolge verwendet und muss zur Begrenzung dieser Zeichenfolge erneut eingegeben werden.

Fehler im Suchbegriff

Diese Meldung erfolgt, wenn in einem CHANGE- oder LOCATE-Kommando der Suchbegriff nicht korrekt angegeben wurde. Die Begrenzer der Zeichenfolge fehlen. Das Kommando ist zu wiederholen. Siehe auch Erlaeuterung der vorhergehenden Meldung.

Tab - Einstellungsfehler

Diese Meldung erfolgt, wenn die aufsteigende Folge bei der Tabulatoreinstellung nicht eingehalten wurde bzw. eine Tabulatorposition groesser 255 war.

Dateieroeffnungsfehler

Diese Meldung erfolgt, wenn ein nicht zugelassener DCP-Dateiname eingegeben wurde, bzw. das Laufwerk die Diskette nicht verarbeiten kann oder eine unformatierte Diskette in das Laufwerk eingelegt wurde.

Schreibfehler Datei

Diese Meldung erfolgt, wenn eine Datei nicht komplett auf die Diskette geschrieben werden kann. Die Ursache kann sein:

- die Diskette ist voll
- ein Schreibfehler
- Laufwerk ist nicht bereit.

Die Datei ist auf eine andere Diskette auszugeben oder ein anderes Laufwerk ist zu verwenden.

Datei nicht gefunden

Diese Meldung erfolgt, wenn in einem ERASE- oder RENAME-Kommando die spezifizierte Datei nicht im aktuellen Laufwerk gefunden werden konnte.

Falsche Funktion

Diese Meldung erfolgt, wenn ein nicht zugelassener Funktionsname in einer DEFINE-Anweisung oder in einer Makroanweisung verwendet wurde. Siehe Abschnitt 5 fuer das Einstellen der Funktionen!

Falsche Taste

Diese Meldung erfolgt, wenn ein nicht zugelassener Tastenname in einer DEFINE- oder Makroanweisung eingegeben wurde. Siehe Abschnitt 5 fuer die Tastenbezeichnung!

Taste ?

Diese Meldung erfolgt, wenn eine Taste bedient wurde, fuer die es keine Funktionszuordnung gibt.

Zeilenmarkierung fehlt

Diese Meldung erfolgt, wenn die [REFLOW]-Funktion aufgerufen wurde und der neu zu formatierende Bereich nicht mit Zeilenmarkierungen versehen war. Fuer diese Funktion sind nur Zeilenmarkierungen zugelassen. Alte Markierungen sind mit der [UNMARK]-Funktion zu loeschen und die neuen Markierungen mit der [MARK LINE]-Funktion einzustellen.

Zeilenueberlauf

Der Editor loescht alle Zeichen, die bei der [JOIN]-Funktion ueber die Position 254 hinaus gehen. Die interne Datei .UNNAMED ist zum Wiederherstellen der Zeile zu verwenden.

Makro Zeilenueberlauf

Diese Meldung erfolgt, wenn in einem DEFINE-Kommando oder in einem Makro eine Definitionszeile zu lang ist. Es sind maximal 120 Zeichen zugelassen.

Block ist bereits markiert

Diese Meldung erfolgt, wenn bereits ein Bereich markiert wurde und erneut eine Markierung eingestellt werden soll. Die alte Markierung ist erst mit der [UNMARK]-Funktion zu loeschen, bevor eine neue Markierung eingestellt werden kann.

Maximale TMP-Dateigroesse erreicht

Diese Meldung erfolgt, wenn die TMP-Datei groesser als 128 KB wird. Die Dateien sind aus dem Speicher auszulagern, um die Laenge zu verkleinern.

Speicher voll - Aufruf beendet

Diese Meldung erfolgt, wenn der Speicherplatz nicht mehr ausreicht. Dateien sind auszulagern bzw. abzuschliessen, um Platz zu schaffen. Eine Kontrolle des zur Verfuegung stehenden Speichers ist mittels des QUESTION MARK-Kommandos vorzunehmen.

Speicher voll - entferne Dateien

Diese Meldung erfolgt, wenn beim Einlesen der aufgerufenen Datei nicht genug Speicherplatz zur Verfuegung steht, um die Datei komplett aufzunehmen. Andere aktive Dateien sind mit dem QUIT- oder FILE-Kommando zu entfernen, um Platz im Speicher zu schaffen. Das Zeichen "@"in der Statuszeile informiert, ob noch Dateien aus dem Speicher auszulagern sind. Erst, wenn dieses Zeichen vom Bildschirm verschwunden ist, reicht der Speicherplatz aus, um die Datei aufzunehmen.

Dateiname fehlt

Diese Meldung erfolgt, wenn beim SAVE- oder FILE-Kommando eine Datei ohne Namen angegeben wurde bzw. im NAME-Kommando das Leerzeichen zwischen Kommando und Dateinamen vergessen wurde.

Es fehlt ' oder]

Diese Meldung erfolgt, wenn in einer DEFINE-Zuweisung das "'" bei einer Literalzuweisung bzw. die schliessende "]" fuer einen Funktionsabschluss fehlt.

Monochrombildschirm nicht installiert

Diese Meldung erfolgt, wenn mit dem SET DISPLAY-Kommando ein Monochrombildschirm zugewiesen wurde, aber dieser am Geraet nicht angeschlossen ist. Diese Einstellung ist fuer den angeschlossenen Bildschirm vorzunehmen.

Name bereits im Zugriff

Diese Meldung erfolgt, wenn mit dem NAME-Kommando der aktuellen Datei der Name einer bereits aktiven Datei zugewiesen wird. Es ist ein anderer Name zu verwenden.

Neue Datei

Diese Meldung erscheint, wenn eine Datei mit dem EDIT-Kommando zur Bearbeitung aufgerufen wird und diese Datei noch nicht im aktuellen Verzeichnis enthalten ist.

= fehlt in Definition

Diese Meldung erfolgt, wenn bei Funktionszuordnungen zu den Tasten das Zuweisungszeichen "=" vergessen wurde. Die Einstellung ist zu korrigieren.

Kein Austausch

Diese Meldung erfolgt bei der Nichtausfuehrung der Funktion [confirm change] in einem vorher aufgerufenen CHANGE-Kommando. Diese Meldung ist kein Fehler, sondern nur ein Hinweis.

Keine Makrodefinition

Diese Meldung erfolgt, wenn in einem DEFINE-Kommando die Reihenfolge der Zuweisungen nicht eingehalten wurde. Das Kommando erwartet zur Zuweisung einen Tastennamen, danach das Zuweisungszeichen und im Anschluss eine Funktion oder ein Literal und eine Funktion.

Markierung fehlt

Diese Meldung erfolgt, wenn eine Funktion aufgerufen wurde, die auf einen markierten Bereich wirkt, aber kein markierter Bereich eingestellt wurde.

Kein Drucker; r/c (Wiederh./Abbruch)

Diese Meldung erfolgt, wenn eine Datei gedruckt werden soll, aber kein Drucker angeschlossen bzw. nicht bereit ist. Durch Eingabe von "r" erfolgt die Wiederholung der Druckausgabe. Die Eingabe von "c" bewirkt den Abbruch des Druckkommandos.

Zu wenig Speicher - druecke eine Taste

Diese Meldung erscheint, wenn nicht ausreichend Speicherplatz zur Verfuegung steht, um eine aufgerufene Operation auszufueh-

ren. Durch Druecken einer Taste wird wieder zum Systemgrundzustand zurueckgekehrt.

Nicht gefunden

Diese Meldung erscheint, wenn nach einem LOCATE- oder CHANGE-Kommando der Suchbegriff nicht mehr gefunden wird, bzw. wenn Absaetze umformatiert werden sollen und das Ende der Datei erreicht wurde.

SET Parameter fehlen

Diese Meldung erscheint, wenn ein SET-Kommando eingegeben wurde, aber die Liste der zu setzenden Parameter fehlt bzw. unvollstaendig ist.

Fehler Quelle und Ziel

Diese Meldung erfolgt, wenn ein markierter Bereich kopiert oder verschoben werden soll und sich Quell- und Zielbereich ueberlappen. Das ist nicht zugelassen.

TMP-Disk ist voll

Diese Meldung erscheint, wenn die TMP-Datei nicht mehr komplett auf die Diskette ausgegeben werden kann. Die maximale Laenge dieser Datei kann 128 K Byte betragen. Es sind unter Verwendung der Kommandos QUIT oder FILE Dateien abzuschliessen, um Platz im Speicher zu schaffen und damit die Groesse der TMP-Datei zu verkleinern. Diese Meldung erfolgt auch, wenn die aktive Datei geladen wird, aber nicht genug Speicherplatz zur Verfuegung steht, um die Datei in den Speicher aufzunehmen.

TMP Fehler; r/c (Wiederh./Abbruch)

Diese Meldung erfolgt, wenn eine TMP-Datei auf dem aktuellen Laufwerk angelegt werden soll, die Diskette aber schreibgeschuetzt ist bzw. bereits eine TMP-Datei enthaelt. Die Diskette ist zu wechseln und "r" einzugeben, um einen neuen Versuch zu starten. Die Eingabe von "c" bricht das Kommando ab.

TMP-Datei angelegt

Dieser Hinweis erfolgt, wenn eine TMP-Datei (BE.TMP) auf die Diskette erfolgreich ausgegeben wurde.

Letzte TMP Zeilen - Leer ersetzt

Diese Meldung erfolgt, wenn beim Schreiben der TMP-Datei ein Schreibfehler auftritt. Es werden dann Leerzeichen ausgegeben und die Ausgabe abgebrochen. Die letzte vorherige abgespeicherte Dateiversion ist fuer die Weiterarbeit zu verwenden.

Zu viele Dateien

Diese Meldung erfolgt, wenn mehr als 20 Dateien aktiviert werden. Mittels Kommando QUIT oder FILE ist die Anzahl der aktiven Dateien zu verringern.

Zu viele Tabs

Diese Meldung erfolgt, wenn beim SET TABS-Kommando mehr als 20 Tabulatorpositionen eingegeben wurden. Die Anzahl ist einzugrenzen.

Druecke ein Zeichen

Dieser Hinweis erfolgt, wenn die Funktionen [FILL MARK] oder [ESCAPE] aufgerufen wurde. Ein Hinweis, dass der Editor nun die Eingabe eines Zeichens von der Tastatur erwartet.

Parameterfehler

Diese Meldung erfolgt, wenn die Wahlparameter "KEY", "MARGINS", "TABS" oder "MEMORY" nicht korrekt eingegeben wurden.

Kommandofehler

Diese Meldung erfolgt, wenn in der Befehlszeile ein unkorrektes Kommando eingegeben wurde.

Schreibgeschuetzt

Diese Meldung erscheint, wenn eine Datei gespeichert werden soll, die Diskette aber schreibgeschuetzt ist.

Anhang

Uebersicht der Tastaturzuordnung fuer Kursorbewegungen und der Arbeit mit Markierungen

Uebersicht ueber Kursortastenbewegungen

Taste	Kursorfunktion
< ↑ >	Kursor 1 Zeile nach oben
< ↓ >	Kursor 1 Zeile nach unten
< <- >	Kursor 1 Zeichen nach links
< -> >	Kursor 1 Zeichen nach rechts
<CTRL>+< <- >	Kursor 40 Zeichen nach links
<CTRL>+< -> >	Kursor 40 Zeichen nach rechts
< 'Y >	Kursor auf 1.Zeichen in der Zeile
<END>	Kursor nach letztes Zeichen auf der Zeile
<CTRL>+< 'Y >	Kursor auf 1. Zeile in der Datei
<CTRL>+<END>	Kursor auf letzte Zeile in der Datei
<PAGE UP>	Blaettern eine Seite nach oben
<PAGE DOWN>	Blaettern eine Seite nach unten
<CTRL>+<PAGE UP>	Kursor 1. Zeile des Bildschirmes
<CTRL>+<PAGE DOWN>	Kursor auf letzte Zeile des Bildschirmes

Uebersicht ueber Markierungen und Stellung des Kursors

Operation	Markierungsart		
	Zeilen ALT+L	Block ALT+B	Zeichen ALT+C
Kopieren ALT+Z	Zeile danach	auf Zeile	auf Zeile
Loeschen ALT+D	beliebige Position	beliebige Position	beliebige Position
Ueberlagerung ALT+O	-	linke obere Position	-
Verschieben ALT+M	Zeile danach	linke obere Position	linke obere Position
ALT+U	loescht alle Markierungen		

IV. BIBLIOTHEKSVERWALTER LIB

1. Einleitung

Der Bibliotheksverwalter LIB erstellt und pflegt Programmbibliotheken, die aus einem oder mehreren Objektmodulen bestehen. Objektmodule sind assemblierte oder compilierte Befehle und Daten. Eine Bibliothek speichert die Objektmodule, die von anderen Programmen zur Ausfuehrung benoetigt werden. Sie wird vom Programmbinder (LINK) zum Einfuegen der Routinen und Variablen verwendet, die nicht im Quellcode des Programmes definiert sind.

LIB erstellt eine Bibliothek durch Kopieren des Inhaltes einer oder mehrerer Objektdateien in die Bibliotheksdatei. Eine Objektdatei enthaelt einen Objektmodul, der durch den Makroassembler MASM oder durch einen Sprachcompiler einer hoeheren Programmiersprache erstellt wurde.

Fuegt LIB einen Objektmodul zu einer Bibliothek hinzu, wird der Modulname in das Verzeichnis der Bibliothek geschrieben. Sucht LINK in der Bibliothek nach den Namen der Routinen und Variablen, die fuer ein Programm benoetigt werden, wird das Verzeichnis der Bibliothek gelesen. Ist die Routine enthalten, wird eine Kopie des Inhaltes dieses Moduls zum Programm gebunden.

Die Aufgaben des LIB sind:

- Erstellen einer neuen Bibliothek
- Pflege einer Bibliothek
- Pruefen des Inhaltes eine Bibliothek
- Erstellen einer Bibliotheks-Referenz-Datei

Das Pflegen einer Bibliothek wird am haeufigsten verwendet, folgende Befehlssymbole sind dazu notwendig:

Symbol	Bedeutung
+	Hinzufuegen eines Moduls
-	Loeschen eines Moduls
-+	Ersetzen eines Moduls
*	Kopieren eines Moduls
-*	Transport eines Moduls

Die Befehle zum LIB koennen eingegeben werden:

- nach Prompts (Eingabeaufforderung),
- in einer Befehlszeile,
- in einer Antwortdatei oder
- mit einer Kombination der drei Methoden.

2. Starten von LIB

Der Name der zu bearbeitenden Bibliothek sowie Befehle, die spezifizieren, was LIB ausfuehren soll, sind einzugeben.

2.1. Starten mit Prompts

- Eingabe **LIB**

  LIB startet und zeigt die Aufforderung an:

  Library name:

- Eingabe **<bibliotheksname>** der Bibliothek, mit der gearbeitet werden soll.
  Wird keine Dateierweiterung eingegeben, fuegt LIB die Erweiterung .LIB an. LIB sucht die spezifizierte Datei. Wird sie gefunden, zeigt LIB das naechste Prompt an, wenn nicht, erfolgt die Anzeige:

  Library file does not exist. Create?
  (Bibliotheksdatei existiert nicht. Erstellen?)

  Eingabe Y: Datei erstellen
  Eingabe N: Beenden LIB

  Soll die Standard-Seitengroesse geaendert werden, kann der Schalter **/PAGESIZE:<zahl>** nach dem Bibliotheksnamen eingegeben werden. <zahl> ist die gewuenschte Seitengroesse. Danach erfolgt die Anzeige:

  Operations:

- Eingabe **<befehle>**

  (Modul Loeschen, Hinzufuegen, Ersetzen, Kopieren, Transportieren)
  Sind mehr Befehle erforderlich, als auf eine Zeile passen, ist & als letztes Zeichen der Zeile einzugeben und ENTER zu bedienen. Danach wird **"Operations"** fuer weitere Eingaben angezeigt. Sind alle Befehle eingegeben, ist ENTER zu bedienen. Das naechste Prompt wird angezeigt:

  List file:

- Eingabe **<protokolldateiname>** (Bibliotheks-Referenzdatei)

  Wenn gewuenscht, ist die Dateierweiterung mit einzugeben, LIB legt hier keine Dateierweiterung fest. Soll diese Datei nicht erstellt werden, ist hier nur ENTER einzugeben.
  Wurde bei "Operations" kein Befehl eingegeben, erstellt LIB die Protokolldatei und beendet das Programm. Anderenfalls zeigt LIB die naechste Eingabeaufforderung an:

Output library:

- Eingabe **<neuer bibliotheksname>**

  Wird keine Dateierweiterung eingegeben, haengt LIB die Erweiterung .LIB an. Soll der Name der bestehenden Bibliotheksdatei nicht geaendert werden, ist nur ENTER zu bedienen. In diesem Fall erstellt LIB eine "Backup-Kopie" der alten Bibliothek durch Ersetzen der Dateierweiterung .LIB durch .BAK.

LIB fuehrt nun die eingegebenen Befehle aus.

Man kann ein Semikolon nach jeder Eingabe (ausser nach LIB) als Standardantwort fuer die restlichen Eingaben verwenden. Dabei wird keine Protokolldatei erstellt, die modifizierte Bibliothek wird unter dem gleichen Namen abgelegt.

Beispiel:

```
Library name: MAT
Operations: +CHAR +TEST
List file: MAT.LST
Output library: MAT1
```

Aus der Bibliothek MAT.LIB, den Dateien CHAR.OBJ und TEST.OBJ wird die neue Bibliothek MAT1.LIB und die Protokolldatei MAT.LST erstellt. Die Bibliothek MAT.LIB bleibt unveraendert.

2.2. Starten mit Befehlszeile

LIB kann auch gestartet werden, indem alle Befehle und Dateien in einer Befehlszeile aufgefuehrt werden.

Format:

LIB <bibliotheksname>[/PAGESIZE:<zahl>][<befehle>][,[<protokolldateiname>][,[<neuer bibliotheksname>]]][;]

<bibliotheksname>	Bibliotheksdatei, mit der gearbeitet werden soll. Wird keine Dateierweiterung eingegeben, fuegt LIB die Erweiterung .LIB an.
/PAGESIZE:<zahl>	Definiert die Seitengroesse der Bibliothek. Standard: 16 Bytes
<befehle>	Spezifizieren der auszufuehrenden Operationen (Modul Loeschen, Hinzufuegen, Ersetzen, Kopieren, Transportieren),
<protokolldatei-name>	Name der Bibliotheks-Referenz-Datei (Protokolldatei). Wird kein Name eingegeben, wird diese Datei nicht erstellt.

<neuer Bibliotheksname>	Name fuer die modifzierte Bibliotheksdatei, wird kein Name eingegeben, verwendet LIB den <bibliotheksnamen> und bezeichnet die alte Bibliothek mit der Dateierweiterung .BAK.

Befindet sich eine der spezifizierten Dateien in einem anderen Verzeichnis oder anderen Laufwerk oder soll sie dorthin geschrieben werden, dann ist der entsprechende Pfadname vor den Dateinamen zu schreiben.
Soll eine Protokolldatei erstellt werden, ist diese vom letzten Befehl durch ein Komma zu trennen.
Wird ein neuer Bibliotheksname eingegeben, ist dieser von der Protokolldatei ebenfalls durch ein Komma zu trennen bzw. vom letzten Befehl durch zwei Kommas.
Man kann ein Semikolon nach jeder Eingabe (ausser nach LIB) als Standardantwort fuer die restlichen Eingaben verwenden. Das Semikolon muss das letzte Zeichen auf der Zeile sein. Standardmaessig wird keine Protokolldatei erstellt, die modifizierte Bibliothek wird unter dem gleichen Namen abgelegt.

Beispiele:

```
LIB MAT +MULT
```

Die Datei MULT.OBJ wird zur Bibliothek MAT.LIB hinzugefuegt, es wird keine Protololldatei erstellt, die Erweiterung wird unter dem gleichen Dateinamen aufgezeichnet und die alte Datei in MAT.BAK umbenannt.

```
LIB MAT +DIV,MAT1.LST,MAT1
```

Die Bibliothek MAT1.LIB wird aus dem Inhalt der Dateien MAT.LIB und DIV.OBJ erstellt, eine Protokolldatei MAT1.LST wird aufgezeichnet, die Datei MAT.LIB bleibt unveraendet.

2.3 Starten mit Antwortdatei

LIB wird durch Lesen der Antwortdatei die die Befehle und Dateinamen enthaelt, abgearbeitet.

Format:

```
LIB @<antwortdatei>
```

Die Antwortdatei kann bei einem beliebigen Prompt oder in einer Befehlszeile spezifiziert werden. Das Starten mit einer Antwortdatei wird genauso behandelt, als haette man zu den Prompts oder in einer Befehlszeile die Eingaben getaetigt.

Zu beachten ist:

- ENTER in der Antwortdatei entspricht ENTER bei einem Prompt bzw. Komma in der Befehlszeile.

- <antwortdatei> muss exakt die Bezeichnung der Antwortdatei

sein. Befindet sich diese Datei in einem anderen Laufwerk oder Verzeichnis, sind Laufwerks- und Pfadname mit einzugeben.

- Die Antwortdatei kann beliebig bezeichnet werden.

Form der Datei:

```
<bibliotheksname>[/PAGESIZE:<zahl>][;]
[<befehle>][;]
[<protokolldateiname>][;]
[<neuer bibliotheksname>]
```

Elemente, die bereits durch Prompts oder eine Teilbefehlszeile bestimmt sind, sind wegzulassen.

Jeder Dateiname muss auf einer separaten Zeile erscheinen. Mehrere Befehle koennen in einer Zeile eingegeben werden. Sind mehr Befehle notwendig, als auf eine Zeile passen, ist & am Zeilenende einzugeben.

Das Semikolon kann auf eine beliebige Zeile gesetzt werden, danach werden dann die Standardantworten eingesetzt. Der Rest der Datei wird ignoriert.

Wird mit Antwortdatei gearbeitet, zeigt LIB jede Antwort aus der Datei auf dem Bildschirm in Form der Prompts an.

Enthaelt die Antwortdatei nicht die notwendigen Eingaben, fordert LIB die fehlenden Angaben und wartet auf eine Tastatureingabe.

Die Antwortdatei muss mit Semikolon oder ENTER enden. Fehlt diese Endebedingung, zeigt LIB die letzte Zeile der Antwortdatei an und wartet, bis ENTER bedient wird.

Beispiel:

```
LIB @BSP

Inhalt BSP: MAT2
            +DIV +UP
            MAT2.LST;
```

Zur Datei MAT2.LIB werden DIV.OBJ und UP.OBJ hinzugefuegt, die Protokolldatei MAT2.LST wird erstellt.

2.4. Setzen Schalter "PAGESIZE"

Durch Hinzufuegen des Schalters "/PAGESIZE" nach dem Bibliotheksnamen kann die Seitengroesse der Bibliothek geaendert werden.

Format:

/PAGESIZE:<zahl> oder **/P:<zahl>**

<zahl>: Spezifiziert die neue Seitengroesse,
= Potenz von 2 zwischen 16 und 32768

Die Seitengroesse einer Bibliothek beeinflusst die Stellung der Module, wie sie in der Bibliothek gespeichert sind.
Die Module beginnen immer am Anfang eines Bereiches, der ein Vielfaches der Seitengroesse (in Bytes) ist, berechnet vom Anfang der Datei.
Die Seitengroesse fuer eine neue Bibliothek ist standardmaessig 16 Bytes bzw. fuer eine existierende Bibliothek die zuletzt definierte Groesse.
LIB verwendet die Indextechnik zum Suchen der Module in der Bibliothek.

Fuer jeden Modul einer Bibliothek geht durchschnittlich eine halbe Seitengroesse Speicherplatz verloren.

Beispiel:

```
LIB TAB/PAGESIZE:256 +EXP +ZINS;
oder
LIB TAB/P:256 +EXP +ZINS;
```

Eine Bibliothek TAB.LIB wird aus den Dateien EXP.OBJ und ZINS.OBJ mit einer Seitengroesse von 256 erstellt.

3. Schaffen einer neuen Bibliothek

Die Befehlseingaben fuer eine neue Bibliothek erfolgen genauso als ob eine vorhandene Bibliothek modifiziert werden soll. Nach der Eingabe des Namens sucht LIB nach dieser Datei. Existiert die Bibliotheksdatei nicht, erscheint die Anzeige:

Library file does not exist. Create?
(Bibliotheksdatei existiert nicht. Erstellen?)

Eingabe: Y: Datei eroeffnen
Eingabe: N: Beenden LIB

LIB erstellt die neue Bibliothek und fordert zur naechsten Befehlseingabe auf bzw. fuehrt die bereits eingegebenen Befehle aus.

4. Pflege einer Bibliothek

Bibliotheken koennen durch Hinzufuegen, Loeschen, Ersetzen und/oder Transportieren von Modulen veraendert werden. Die hierfuer notwendigen Befehle koennen bei der Eingabeaufforderung "Operations:", in der Befehlszeile bei <befehle> oder in einer Antwortdatei eingegeben werden.
Vor dem Start von LIB ist abzusichern, dass auf der Diskette genuegend Platz fuer weitere Dateien (Protokolldatei, neue Bibliotheksdatei) vorhanden ist. Reicht der Platz nicht aus, wird

eine Fehlermeldung angezeigt.

4.1. Hinzufuegen eines Moduls

Syntax:

+<objektdatei>

Dieser Befehl fuegt den Objektmodul der spezifizierten <objektdatei> an die aktuelle Bibliothek an. Wird keine Dateierweiterung spezifiziert, ergaenzt LIB den Standard .OBJ. Befindet sich die Datei in einem anderen Laufwerk bzw. in einem anderen Verzeichnis ist der gueltige Pfadname zu ergaenzen.
Es sind **keine** Leerzeichen zwischen dem Pluszeichen und dem Dateinamen einzugeben.
LIB sucht nach der bezeichneten Datei und fuegt den Inhalt der Objektdatei an das Ende der aktuellen Datei an. Danach entfernt LIB Laufwerkname, Pfadname (wenn definiert) und die Dateierweiterung und schreibt den Rest des Namens in die Inhaltstabelle der Bibliothek.

Beispiel:

LIB MAT +A:\TAB\SIN;

SIN.OBJ wird zur Bibliothek MAT.LIB hinzugefuegt, die Objektdatei befindet sich im \TAB-Verzeichnis im Laufwerk A. Die alte Bibliothek MAT.LIB wird in MAT.BAK umbenannt, es wird keine Protokolldatei erstellt.

4.2. Loeschen eines Moduls

Syntax:

-<modulname>

Ein Objektmodul mit der Bezeichnung <modulname> wird aus der aktuellen Bibliothek geloescht.
LIB fuehrt zuerst alle Loeschbefehle aus bevor Module hinzugefuegt werden. Dies ist unabhaengig von der Eingabeanordnung der Befehle. Soll eine neue Version eines Moduls eine bereits in der Bibliothek vorhandene ersetzen, wird durch diese Reihenfolge der Ausfuehrung eine Verwechslung verhindert.

Beispiel:

LIB MAT -COS,,MULT;

Die Bibliothek MULT.LIB wird durch Kopieren des Inhaltes von MAT.LIB ohne COS.OBJ erstellt.

4.3. Ersetzen eines Moduls

Syntax:

-+<modulname>

Der durch <modulname> bezeichnete Modul wird durch eine Objektdatei gleichen Namens ersetzt. LIB loescht zuerst den Modul. Dann sucht es im aktuellen Verzeichnis nach einer Datei mit gleichem Namen und der Dateierweiterung .OBJ und fuegt diese Datei an das Ende der aktuellen Bibliothek an. Wird der zu ersetzende Modul oder die Objektdatei nicht gefunden, erfolgt eine Fehlermeldung.

Beispiel:

LIB MAT -+TAN,MAT.LST;

Der Modul TAN wird in der Bibliothek geloescht, die Objektdatei TAN.OBJ im aktuellen Verzeichnis gesucht und zum Inhalt der Bibliothek hinzugefuegt. Eine Protokolldatei MAT.LST wird erstellt.

4.4. Kopieren eines Moduls

Syntax:

***<modulname>**

Der Befehl kopiert einen mit <modulname> benannten Modul aus der Bibliothek und schreibt ihn als eine Objektdatei mit gleichem Namen und der Dateierweiterung .OBJ in das aktuelle Verzeichnis. Befindet sich der Modul nicht in der Bibliothek, erfolgt eine Fehlermeldung.

Beispiel:

LIB MAT *ALG1;

Eine Datei mit der Bezeichnung ALG1.OBJ, die aus dem Modul ALG1 besteht, wird im aktuellen Verzeichnis abgelegt. Die Bibliothek MAT.LIB bleibt unveraendert.

4.5. Transport eines Moduls

Syntax:

-*<modulname>

Der Transportbefehl schreibt den mit <modulname> benannten Modul als Objektdatei gleichen Namens mit der Erweiterung .OBJ ins aktuelle Verzeichnis und loescht ihn in der Bibliothek. Befindet sich der Modul nicht in der Bibliothek, wird eine Fehlermeldung angezeigt.

Beispiel:

```
LIB MAT -*ALG2;
```

Der Modul ALG2 wird in der Bibliothek MAT.LIB geloescht und als Objektdatei ALG2.OBJ im aktuellen Verzeichnis abgelegt.

4.6. Verbinden von Bibliotheken

Syntax:

```
+<bibliotheksname>.LIB
```

Der Anfuegebefehl kann auch verwendet werden, um den Inhalt einer anderen Bibliothek zur aktuellen Bibliothek hinzuzufuegen. <bibliotheksname> ist der Name der Bibliotheksdatei, die angefuegt werden soll.
Die Dateierweiterung ist anzugeben. Anderenfalls sucht LIB eine Objektdatei dieses Namens.
LIB haengt die Module der bezeichneten Bibliothek an das Ende der aktuellen Bibliothek an, ohne die benannte Bibliothek zu zerstoeren bzw. die Module zu loeschen.

Beispiel:

```
LIB MAT +TABELLE.LIB
```

Die Module der Bibliothek TABELLE.LIB werden zur Bibliothek MAT.LIB hinzugefuegt.

5. Pruefen des Inhaltes einer Bibliothek

Der Inhalt einer Bibliothek kann geprueft werden, um das ordnungsgemaesse Abarbeiten von LINK zu abzusichern.

Format:

```
LIB <bibliotheksname>;
```

Bei der zeilenweisen Eingabe ist ein Semikolon nach dem Bibliotheksnamen bzw. bei "Operations" einzugeben.
Bei fehlerhafter Bibliothek zeigt LIB die entsprechende Meldung an. Diese Pruefung kann nach dem Kopieren einer Bibliothek zur Sicherheit durchgefuehrt werden.

Beispiel:

```
LIB MAT;
```

Die Bibliothek MAT.LIB wird geprueft.

6. Erstellen einer Bibliotheks-Referenz-Datei (Protokolldatei)

LIB erstellt eine Bibliotheks-Referenz-Datei, wenn beim Prompt "List file" oder in der Befehlzeile an der Position <protokoll-dateiname> ein Dateiname eingegeben wird.
Diese Protokolldatei besteht aus zwei Listen, einer Liste aller PUBLIK-Symbole, die sich in der Bibliothek befinden, und einer Liste aller Module der Bibliothek.

Liste 1 - PUBLIKSYMBOLE:
Alle Symbole werden alphabetisch geordnet aufgefuehrt, jedem Symbolnamen folgt der Name des Moduls, in dem das Symbol vorkommt.

Liste 2 - MODULE:
Alle Module werden in der Reihenfolge aufgefuehrt, wie sie in der Bibliothek stehen, dem Modulnamen folgen die PUBLIK-Symbole, die in dem jeweiligen Modul vorkommen.

Beispiel:

```
LIB
Library name: anz
Operations:
List file: anz;
```

Die Bibliothek ANZ.LIB besteht aus den Modulen char, test und moni. Es wird eine Protokolldatei mit der Bezeichnung ANZ mit folgendem Inhalt erstellt:

PUBLIKSYMBOLE:

```
CALC_POINTER......moni          CHECK.............moni
CHECK_PARM........moni          DISPLAY...........moni
DO_IT.............moni          INIT_SCREEN.......moni
MAIN..............moni          NEXT..............moni
NO_PARM...........moni          READ_PARM.........moni
RETURN............moni          STORE_PARM........moni
TEST_PARM.........moni          VALID_CHECK.......moni
```

MODULE:

```
char            Offset: 00000010H  Code and data size: 106H

test            Offset: 00000050H  Code and data size: 108H

moni            Offset: 000000b0H  Code and data size: 659H
  CALC_POINTER    CHECK            CHECK_PARM       DISPLAY
  DO_IT           INIT_SCREEN      MAIN             NEXT
  NO_PARM         READ_PARM        RETURN           STORE_PARM
  TEST_PARM       VALID_CHECK
```

Die Protokolldatei kann mit folgender Zeile auf den Bildschirm ausgegeben werden, anstatt sie auf Diskette aufzuzeichnen:

```
LIB anz,con;
```

7. Fehleranzeigen

Vom Programm LIB koennen folgende Fehlermeldungen angezeigt werden:

filename: cannot acces file
filename ist kein gueltiger Objektmodul.

** Error: cannot create extract file **filename**; cannot continue
Status: 1
Die Diskette oder das Stammverzeichnis ist voll, oder die durch **filename** bezeichnete Datei existiert bereits mit dem Attribut "read only". Auf der Diskette ist Platz zu schaffen bzw. das Dateiattribut zu aendern.

** Error: cannot create new library; cannot continue
Status: 1
Die Diskette oder das Stammverzeichnis ist voll, oder die Bibliotheksdatei existiert bereits mit dem Attribut "read only". Auf der Diskette ist Platz zu schaffen bzw. das Attribut "read only" zu aendern.

** Error: cannot create listing; cannot continue
Status: 1
Die Diskette oder das Stammverzeichnis ist voll. Auf der Diskette ist Platz zu schaffen.

** Error: cannot open response file; cannot continue
Status: 1
Die angegebene Antwortdatei wurde nicht gefunden.

cannot open VM.TMP
Die Diskette oder das Verzeichnis ist voll, Dateien loeschen oder auf andere Diskette schreiben, um Platz zu schaffen.

cannot read from VM
Programmfehler

** Error: cannot rename old library; cannot continue
Status: 1
LIB konnte die alte Bibliotheksdatei nicht in die Dateierweiterung .BAK umbenennen, weil eine .BAK-Version bereits mit dem "read only" Attribut existiert. Das "read only" Attribut der alten .BAK-Version ist zu aendern.

cannot reopen library
Die alte Bibliotheksdatei konnte nicht wiedereroeffnet werden, nachdem sie in eine .BAK-Erweiterung umbenannt wurde.

cannot write to VM
Programmfehler

** Error: comma or newline expected; cannot continue
Status: 1
Ein Komma oder ENTER wurde in der Befehlszeile an falscher Stelle eingegeben bzw. ein Komma wurde vergessen einzugeben.
Z.B. LIB MAT, -DIV +POT;
Die Zeile muss richtig heissen:
LIB MAT -DIV +POT;

** Error: error writing to new library; cannot continue
Status: 1
Die Diskette oder das Stammverzeichnis ist voll, Dateien loeschen oder auf andere Diskette schreiben, um Platz zu schaffen.

** Error: error writing to cross referece file; cannot continue
Status: 1
Die Diskette oder das Stammverzeichnis ist voll, Dateien loeschen oder auf andere Diskette schreiben, um Platz zu schaffen.

free: not allocated
Programmfehler

insufficient memory
LIB hat nicht genug Speicher zur Verfuegung, einige Shells oder residente Programme entfernen oder mehr Speicher hinzufuegen.

internal failure
Programmfehler

filename: invalid library header
Die Bibliothek stimmt nicht mit dem Format ueberein, das von LIB erwartet wird.

** Error: invalid library, cannot continue
Status: 1
Die Bibliothek stimmt nicht mit dem Format ueberein, das von LIB erwartet wird.

invalid object module **name** near **location**
in file **libraryname**
Der durch Name spezifizierte Modul ist kein gueltiger Objektmodul.

mark: not allocated
Programmfehler

missing terminator
Die Antwort zu einer Ausgabebibliothek wurde nicht mit ENTER beendet.

filename: modul not in library
Der spezifizierte Modul befindet sich nicht in der Bibliothek.

no more virtual memory
Programmfehler

xxx: page size too smal--ignored
Die mit /Pagesize spezifizierte Seitengroesse muss 16 oder groesser sein.

filename: modul redefinition in file **filename** ignored.
Der angegebene Modul befindet sich bereits in der Bibliothek.

too many symbols
Die maximale Anzahl der Symbole in einer Bibliothek ist 4609.

** Error: syntax error, cannot continue
Status: 1
Der eingegebene Befehl entspricht nicht der LIB-Syntax.

** Error: syntax error (bad input), cannot continue
Status: 1
Das eingegebene Befehl entspricht nicht der LIB-Syntax.

** Error: syntax error (bad file spec), cannot continue
Status: 1
Ein Befehlsoperator wie z.B. das Minus-Vorzeichen wurde ohne nachfolgenden Modulnamen eingegeben.

** Error: syntax error (switch name expected), cannot continue
Status: 1
Eingabe / ohne Pagesize-Option

** Error: syntax error (switch val expected), cannot continue
Status: 1
Eingabe /Pagesize ohne nachfolgenden Wert

unexpected EOF on command input
Ein Dateiendezeichen wurde vorzeitig als Antwort zu einem Prompt eingegeben.

** Error: unknown switch, cannot continue
Status: 1
Eine falsche Option wurde eingegeben, /Pagesize ist die einzige gueltige Option.

** Error: write to extact file failed, cannot continue
Status: 1
Die Diskette oder das Stammverzeichnis ist voll, Dateien loeschen oder auf andere Diskette schreiben, um Platz zu schaffen.

** Error: write to library file failed, cannot continue
Status: 1
Die Diskette oder das Stammverzeichnis ist voll, Dateien loeschen oder auf andere Diskette schreiben, um Platz zu schaffen.

8. EXIT-Codes

Dieser Code, auch "errorlevel"-Code genannt, kann mit Hilfe von BATCH-Dateien (siehe "Anleitung fuer den Bediener" 14. Stapelverarbeitung) oder MAKE abgefragt werden.

Der Exit-Code Ø zeigt an, dass das Programm ohne Fehler ausgefuehrt wurde.
LIB gibt beim Auftreten von Fehlern verschiedene Codes zurueck, die die unterschiedlichen Fehlerarten anzeigen:

Code	Bedeutung
Ø	kein Fehler
1	LIB-Fehler werden nicht aufgelistet
4	interner Fehler
3	zu viele Symbole

1. 62. 548310. 8. D